STRUCTURAL LEARNING
I. Theory and Research

JOSEPH M. SCANDURA
STRUCTURAL LEARNING

I. Theory and Research
STRUCTURAL LEARNING SERIES

Joseph M. Scandura, Editor

An interdisciplinary series concerned with the following questions: (1) How can competence (knowledge) be characterized in a way which has behavioral relevance? (2) How can human knowledge be operationalized? (3) What are the mechanisms which govern learning? Performance? Motivation? Perception? Memory? (4) How can learning be optimized?

Cognitive, developmental, educational, and mathematical psychologists, philosophers (epistemology), computer scientists (simulation and artificial intelligence, theory of computation), generative linguists, mathematics educators, and those in educational technology will find the series particularly appealing.

1. SCANDURA • Structural Learning: I. Theory and Research

2. SCANDURA (Ed) • Structural Learning: II. Issues and Approaches
   (Contributors: ARBIB • CORCORAN • DOMOTOR • GREENO • LOVELL • NEWELL • ROSENBLOOM • SHAW • SIMON • SUPPES • WITTROCK • WITZ)
STRUCTURAL LEARNING
I. Theory and Research

by

JOSEPH M. SCANDURA

GORDON AND BREACH, SCIENCE PUBLISHERS

NEW YORK • LONDON • PARIS
CONTENTS

PREFACE vii

Chapter 1. INTRODUCTION AND OVERVIEW / 1

1. Overview 2
2. Relation to Traditional Theories 6

Chapter 2. WHAT IS A RULE? 9

1. Rules and Rule-Governed Behavior 9
   1.1 Rule-Governed Behavior 9
   1.2 The Rule Construct 11
2. More Precise Specification 22
   2.1 Characterization of Rule-Governed Behavior 22
   2.2 Characterization of the Rule Mechanism 26
   2.3 Schematic Representation of Procedures and Hierarchies 34
3. Behavior Potential 41
   3.1 Formal Representation of Behavior Potential in Terms of Categories and Functors 43
   3.2 Accounting for Behavior Potential 48

Chapter 3. MATHEMATICAL PRELIMINARIES 51

1. Concrete Embodiments 51
2. Mathematical Systems 56
3. Informal Axiomatic Theories 58
4. Formal Systems 65
5. Proof Theory and Model Theory 68
Chapter 4. RULE-BASED THEORIES OF STRUCTURED KNOWLEDGE:  
GENERAL NATURE AND RELATIONSHIPS TO LINGUISTICS 73
1. Rule Sets and Evaluative Criteria 73
2. Alternative Characterizations 76
3. Relationships to Linguistics 81

Chapter 5. FOUNDATIONS OF THE THEORY OF KNOWLEDGE 87
1. Nature of the Stimuli and Responses 88
2. Rules 91
3. Programs: Descriptions of Rules 100
4. Kinds of Rules 101
5. Nature of the Theory of Knowledge 105
6. Applications 111
7. Axioms, Comments, and Conjectures 114

Chapter 6. A THEORY OF MATHEMATICAL KNOWLEDGE 119
1. Semantic Knowledge - Mathematical Systems and Embodiments 121
   1.1 Simple Rules and Simple Elements 121
   1.2 Relationships Within Systems 124
   1.3 Semantics and Relationships Between Systems and Embodiments 126
   1.4 Behavioral Implications - Instances of Creative Behavior 134
2. Syntactic Knowledge - Formal Systems 141
3. Axiomatics - Relationships Between Semantic and Syntactic Knowledge 145
   3.1 The Nature of (Semantic) Inference Rules 149
   3.2 On Generating New Logical Procedures 157
   3.3 Conjecture-Making, Constructing Counterexamples, Proof and Model Theory 165

Chapter 7. MECHANISMS OF LEARNING AND PERFORMANCE - MEMORY-FREE  
THEORIZING IN STRUCTURAL LEARNING 171
1. Some Preliminaries - Decision Making Capabilities, Goals, and Rules 173
2. Performance Mechanisms - Conditions Under Which (Learned) Rules Are Put to Use 176
3. Assessing Behavior Potential 178
   3.1 Related Research 179
   3.2 The Problem 180
Chapter 7, continued

3.3 A Solution 183
3.4 Testing the Hypothesis 186
3.5 Some Empirical Results 190
3.6 Assessing Behavior Potential 199

4. The Nature of Problem Solving – A Mechanism for Learning 205
   4.1 Basic Hypotheses and Examples 206
   4.2 Experiments on Problem Solving 213

5. Discovery Learning 229

6. Expository Learning 236
   6.1 The Nature of Interpretation 237
   6.2 Properties of Minimal Signs (Symbols and Icons) 242
   6.3 Characterization of Knowledge Underlying Interpretation 245
   6.4 Mechanisms of Learning and Performance in Expository Learning 249
   6.5 On Ambiguity 252

7. Summary and Concluding Remarks 253

Chapter 8. MECHANISMS OF MOTIVATION - MEMORY-FREE THEORIZING IN STRUCTURAL LEARNING 255

1. Nature of Motivation Theory in Structural Learning 256
2. Identifying Motives (Goals) and Rules – Explanation 259
   2.1 Verbal Reports 259
   2.2 Refined Goal Criteria 259
   2.3 Further Comments 262
3. Experience-Free Hypotheses 263
4. Experience-Sensitive Hypotheses 265
   4.1 Mechanisms of Motivation 266
   4.2 Selection Rule Study 271
   4.3 Further Analyses and Implications 275

Chapter 9. FOUNDATIONS OF THE IDEALIZED (MEMORY-FREE) THEORY 285

1. Goal Situations 286
2. Nature of the Idealized (Memory-Free) Theory 287
3. Applications 296
4. Epilogue 298
PREFACE

This book is an indirect outgrowth of an invitational conference on structural learning held at the University of Pennsylvania on April 20 - 22, 1968. At that time a small group of scholars from a number of related fields was brought together in an attempt to forage new ground in a previously virgin territory. We were not interested in reporting finished research but, rather, we were unsatisfied with the kind of research being done on complex human learning and wanted to see what we could do about improving it. More particularly, we wanted to identify some of the basic problems involved and, if possible, come up with new ways of attacking these problems.

As it turned out, the conference was a much greater success than most of us ever dreamed it would be. The interaction—including not less than half-a-hundred spirited arguments—was lively indeed. This writer finds it impossible to capture in words the all-concentrative atmosphere. The clashing red of Corcoran's flushing face and thinning hair as he challenged my formulation of rule-governed behavior or as he reacted to Rosenbloom's reasoned criticism of his discussion of suppositional theories of proof can only be fully appreciated by those who were present. Wittrock's passioned defense of S-R mediation theories and the look in Greeno's eye when Rosenbloom described his mathematical golf were equally memorable. Perhaps the most classic comment to come out of the conference, however, were the two words spoken by R. Duncan Luce. In response to Corcoran's comment that it would take "fifty years to accomplish" something I said, Luce commented in skeptical overtones, "At least!"

Although the chapters included in this book were prepared after the conference, it is safe to say that all were influenced by the ideas expressed there. For some, the conference helped to shape in fundamental
ways the very direction of their future research. Greeno's chapter in Volume II, for example, is a far more vivid account of what he was trying to say at the conference. Corcoran, too, readily acknowledges the value of some of the criticism leveled at his proposals. If it were not for the conference, it is unlikely that I would have even attempted to analyze mathematical structures, let alone attempt to devise a comprehensive theory of structural learning. Without the excellent criticism I received from more than one conferee, I might still be hung up in the unrewarding and seemingly unending cycle of criticizing S-R mediation theory.

Since then three increasingly successful meetings on structural learning have been held and a fourth is being planned as this book goes to press. A number of the contributions to these volumes were initiated at these meetings.

The book is being published in two volumes. In the first volume, *Structural Learning I: The Theory and Empirical Research*, I have described what I see as the rudiments of three deterministic partial theories of structural learning. The first involves competence, partial theories which deal only with the problem of how to account for the various kinds of behavior of which people are typically capable. Special attention is given to mathematical competence. Nothing is said about learning or performance. The second partial theory is concerned with motivation, learning, and performance under idealized conditions, and is obtained from the first partial theory by imposing further structure on it. This theory says nothing about memory or the limited capacity of human subjects to process information. It applies only where the subject has all of the necessary information readily available, and his capacity to process information is essentially unlimited (as when subjects have access to paper and pencil and/or other memory aids). The final theory is obtained from the second by making additional assumptions, which bring memory and finite information processing into the picture. The theory is still partial, however, since no attempt is made to deal with certain ultra-short-term behavioral phenomena which appear to depend directly on particular physiological characteristics (e.g., the short period of time images are presumably retained on the retina of the eye.)

Writing a book such as this is professionally risky. For one thing, it immediately incurs the wrath of specialists whose work has been largely ignored.

A second problem is far more critical and concerns the naivety that is almost certain to creep into a book which attempts to draw from,
synthesize and build upon work in as many different areas as this one. Judged by traditional standards in the various disciplines, certain sections may appear to over-simplify, or even misrepresent (I hope only slightly), some extremely difficult problems. Some mathematicians, for example, would undoubtedly have preferred a more extended formal treatment. In response, I can only ask for understanding as I have tried to make the book both available and useful to as broad an audience as possible. Furthermore, while the formalization of much of what is said is routine, there are other aspects of the proposed theory which may require new kinds of mathematics. The logician, computer scientist, and philosopher of science may feel similarly short changed. As for them, and the linguist, I can only hope that I have not misrepresented their fields too badly -- and, if I have, I hope that my misunderstandings have been creative ones. I feel more secure about my comments concerning psychology since I have spent most of my professional career working in and around the area. Nonetheless, I am sure that many of my colleagues will feel that I have overgeneralized on the basis of only very limited empirical evidence. This is particularly true of my discussion of memory in Chapter 10. My hope here is that psychologists will take up the challenge and help collect more relevant data. Others, particularly those of the neoassociationist persuasion, may find it difficult to accept the conceptualization proposed. The discussion in Chapter 2 is directed largely (but not exclusively) at this group. Still others would probably have preferred a more searching review of the literature in concert with the theoretical development itself. My only defense is that this would have tended to obscure basic issues and is a matter which can reasonably be postponed.

All in all, I hope that the reader will see fit to judge this work in its entirety, and not solely in terms of the pieces. Further development of both a theoretical and empirical nature is needed in just about every sphere, but I am of the opinion that further refinement and extension of the theory (to perception, for example) will be possible without destroying the basic framework. To be sure, further refinement and extension of the theory should be a first order of business. I hope too that the less theoretically inclined will find something of value in the following pages. I am particularly hopeful that psychologists will turn to the task of collecting badly needed empirical data and that educators will begin to build on what appear to me to be some rather clear
cut implications for educational practice.

Structural Learning II: Issues and Approaches is an edited volume and reflects the major approaches being taken in structural learning today. The chapter headings and contributors are: (1) Basic unit in structural learning: Association or automaton (rule)? (Arbib, Scandura, Suppes); (2) New directions for research on rule learning (Scandura, Wittrock); (3) Graph theoretic models (Greeno); (4) Piagetian models (Lovell, Witz); (5) Simulation models (Shaw, Simon and Newell); (6) Competence models in linguistics, logic and mathematics (Corcoran, Domotor, Rosenbloom, Scandura); (7) A theory of structural learning (Scandura).

As is always the case in writing books, much is owed to other people. I want to acknowledge the interesting and often helpful discussions I have had at various stages in the preparation of this book with Michael Arbib, John Corcoran, Zoltan Domotor, James Greeno, Paul Rosenbloom, Robert Shaw, Patrick Suppes, Merlin Wittrock and Klaus Witz, the other contributors, and John Bormuth, Lyle Bourne, John Carr, Fred Davis, Frank Farley, Sol Feferman, Peter Freyd, Robert Gagne, Robert Glaser, Harry Gray, Sol Gorn, Henry Hiz, James Jenkins, Manfred Kochen, Duncan Luce, George Miller, Marvin Minsky, Seymour Papert, George Polya, Dag Prawitz, Joseph Royce, William Rozeboom, and Kellogg Wilson. This does not imply that they would necessarily agree with all that has been said, or indeed with anything that has been said. For that I must assume complete responsibility.

I owe a special debt to many of my students who over the years have prodded me to make good on my contention that there were better ways of dealing with the problems of structural learning and mathematics education. Several, including Louis Ackler, John Durnin, Francine Endicott, Wally Wulfeck and Donald Voorhies have collaborated with me in conducting some of the empirical research reported herein, and this help is gratefully acknowledged. Additional work is underway as this book goes to press and this will be reported in future publications. I would particularly like to thank John Durnin and Wally Wulfeck, John for his help with the Index and Wally for the Bibliography. Tina Baker is partly responsible for proving the manuscript.

The original conferees are indebted to the Graduate School of Education, University of Pennsylvania, for providing partial support and facilities for the original conference on Mathematics and Structural Learning. Morris Viteles and Bill Castetter, former deans, and David
Goddard, former provost, all had a hand in making funds available.

Mary Tye, Peggy Suckle, Roberta Williams, Katherine Whipple, Lee Carvalho, Joan Etchingham and particularly Clara Ueland deserve the credit that is always due typists who labor over draft upon draft of the material before it is completed.

In addition, I am grateful to those agencies and individuals who have provided me with financial and other support during much of my professional career. It is no understatement to say that without this support, this work would not have been possible. I would particularly like to mention the small contracts, graduate research training, and basic research programs of the U.S.O.E. and, more recently, the National Science Foundation. A senior post-doctoral fellowship, supported by the U.S.O.E. research training program, came at an especially helpful time. During this period space and facilities were provided by the University of Pennsylvania, University of California, Berkeley, and Institute for Mathematical Studies in the Social Sciences, Stanford University.

Last, but not least, I would like to express my deep gratitude to my wife, Alice, and children, Jeanne, Janie, Joey and Julie, for their continual support while I struggled to capture and organize ideas that were often as hard to find as the proverbial pot of gold at the foot of a rainbow.

To my parents, I owe the oldest debt, and it is to them that I dedicate this book. Without their many sacrifices, and the confidence they showed in me during my growing years, this book would never have been written.

Joseph M. Scandura
Philadelphia, June 5, 1972
In spite of the diversity which presently exists in behavioral theorizing, reference to probabilistic notions is all-pervasive. Even support at the .05 level of significance is often enough to elicit whoops of glee from most cognitive theorists. Given this milieu, it is not too surprising that (aside perhaps from computer simulation types and a few competence theorists (e.g., Miller and Chomsky, 1963)), no one seems to have seriously pursued the possibility that deterministic theorizing about complex human learning may actually be easier than stochastic theorizing. And yet, this is precisely what in my own work I have found to be the case.

The main purpose of this volume is to describe the "rudiments" of a potentially powerful and internally consistent deterministic theory of structural learning, which could make it possible to explain, predict, and even control (indirectly) certain critical aspects of the behavior of individual subjects in specific situations. The term "rudiments" is used because at the present time relatively few implications of the theory have been drawn out. The emphasis so far has been on establishing a fit between behavioral reality and the basic constructs and hypotheses of the theory.

The to-be-proposed theory consists of three interrelated partial theories, each of which must be tested in a different way. First, there is a theory of structured knowledge—or, more accurately as we shall see, theories of structured competence. These theories deal with the problem of how to characterize knowledge: the knowledge associated with particular behavior constitutes a theory in its own right. The second theory brings the behaving subject into the picture. It provides a basis (1) for determining the knowledge had by particular subjects (relative to a given theory of knowledge) and (2) for telling how that knowledge is selected for use and how new knowledge is acquired. This theory is an idealization in the sense that it applies only where the subject is unencumbered by memory or by his
finite capacity for processing information. The third theory is still more general and tells what happens when memory and information processing capacity are taken into account. These three theories are not independent of one another, although, as we shall see, research on any one can progress independently of the others, and this includes empirical testing.

1. OVERVIEW

Chapter 2 is definitional in nature and addresses itself to the question: What is a rule? Throughout the discussion, a basic distinction is made between rules and rule-governed behavior. Rule-governed behavior is shown to correspond directly to what recursion theorists call partial recursive functions. The abstract notion of a program is generalized to provide a relatively precise language to talk about rules and their component parts. Using this formulation and closely related schematic representations, certain basic relationships between rules, networks of associations, automata, and TOTE hierarchies are made explicit. A distinction is also made between introducing rules to account for observed behavior and theorizing in which it is assumed that human subjects actually learn and use rules. The chapter ends with a discussion of behavior potential, relative to given classes of rule-governed behavior, and in particular, shows how the behavior potential of individual subjects may be characterized in terms of categories and functors.

Chapters 3, 4, 5, and 6 attempt to give psychological reality to such mathematical notions as embodiments, structures, and theories. Certain mathematical preliminaries are provided in Chapter 3 for the behavioral scientist who is not familiar with mathematical foundations. The problem of characterizing structured knowledge is considered in Chapters 4, 5, and 6.

The goal of the theory of knowledge is to find a finite set of rules, and laws which govern their interaction. Given a behavior class of interest, the theorist-observer is obliged to identify a set of rules (his measuring units) and to show how these rules are allowed to interact in accounting for the behavior. In the study of formal systems, for example, rules may be arbitrarily composed. Transformational grammars allow transformations (rules) on phrase structures as well. The proposed theory of knowledge extends this idea by allowing rules to act on other rules at arbitrary levels. (As with other competence theories, of course, each such theory applies only to given classes of potential behavior; one should not expect a theory of knowledge to apply beyond its domain.) In Chapter 4, particular attention is given to general requirements for a theory of (structured) knowledge and relationships to linguistics. In Chapter 5 the foundations of the theory are
both presented more formally and extended to deal with perception and meaning. Chapter 6 deals systematically with the special problems of characterizing mathematical knowledge. Semantics, syntax (formal systems), and relationships between them (axiomatics) are all considered.

Chapters 7, 8, and 9 are concerned with what I have called memory-free theorizing. The idealized theory of structural learning begins where the theory of knowledge leaves off. Such a theory must provide a way to determine which of the rules (or parts thereof) in the rule set (of the theory of knowledge) may be attributed to the subject. We refer to this as "assessing behavior potential." As we shall see, we can only identify the rules had by a subject insofar as they are commensurate with those to which the observer (competence theorist) is sensitized. In a sense, the latter serve as basic measuring units. The idealized theory must also include mechanisms that make it possible to explain and predict behavior in situations where the subject is unencumbered by memory or his limited capacity to process information. Notice, in this case, that the adequacy of such explanations and predictions depends not only on the veridicality of the mechanisms themselves (and the empirical conditions under which a study is run), but also on the prior theory of knowledge. It would not be inaccurate to say that explanation and prediction resulting from the idealized theory is always relative to the adequacy of some given theory of knowledge.

Chapter 7 deals with mechanisms by which rules are put to use and how new rules are acquired under the idealized conditions specified. Empirical support for the proposed mechanisms is reported. These mechanisms are also used as a basis for analyzing the processes of discovery learning and learning by exposition. Chapter 8 deals with mechanisms of motivation, and addresses itself to the problem of explaining and predicting what a subject will do next. This problem is viewed in terms of rule selection—or, why one learned rule is put to use rather than some other. A mathematization of the idealized theory, taken as a whole, is proposed in Chapter 9.

The final chapter (10) adds still more structure to these partial theories by introducing assumptions concerning memory and information processing. This "enriched" theory is obtained from the idealized (memory-free) variety by adding more structure. In order to make predictions in this theory it is necessary to know not only the subject's capacity for processing information and the mechanisms which govern the way information moves between permanent memory and the processor, but also what is in the processor at any given time.

Toward this end: (a) A distinction is made between information (rules) which is actively being processed and information which has been stored in
This distinction corresponds roughly to the widely agreed upon distinction between long and short-term store (memory) although there are some important differences. In the enriched theory, for example, this distinction has as much to do with "arousal" phenomena as with memory.

(b) The mechanisms of the idealized theory are generalized to account for storage and retrieval from permanent memory. (c) Presumed physiologically based limits on the amount of information that can be kept immediately available (or processed) at any one moment are taken explicitly into account. The fixed finite capacity has much in common with Miller's (1956) magic number seven although a number of important extensions and refinements are included.

This theory, nonetheless, is still an idealization in that it does not purport to take still other physiologically based limitations directly into account (although presumably this might be done as well). By such limitations, of course, I am referring to such things as the time available for processing stimulation received from the environment before it is lost. (In other theories this is taken into account by postulating the existence of a sensory register.)

Needless to say, constructing an adequate and operational theory of memory is a complex problem and Chapter 10 is only suggestive as to how this might be accomplished. In particular, rigorous formulation of such a theory depends explicitly on the precise characterization and operationalization of the particular rules a subject is using. In this case, it is no longer sufficient (as it is in the idealized theory) to specify rules up to equivalence classes. They must be known definitively. Fortunately, this is possible to accomplish in principle because rules can be characterized in terms of associations (cf. Chapter 2 and Volume II, Chapter 1). The task of operationalizing these ideas is not simple, however, and many problems still need to be resolved at the time of this writing.

This volume provides a case study of structuralism in process, and is almost exclusively the result of my attempts to make sense of what has too frequently appeared to be an unfathomable quagmire of behavioral evidence, much of it introspective in nature. My approach has been one of looking at the problem of complex human learning and performance as a whole—as a complete structure (cf. Piaget, 1970), and of attempting to conceptualize that whole with increasing rigor through a series of successive approximations.

The mathematically sophisticated reader can get a fairly good idea of the essentials of the theory by reading Chapter 1 together with the following more formal portions of the text, Chapter 5, Chapter 9, Section 2 of Chapter
At the present time many of the major parameters of the evolving theory have been identified; the theory nonetheless is far from finished. Relatively little is said in this volume about perception, and the theory of memory is still in a rudimentary and untested state. This work, then, really constitutes just a beginning; it provides a base upon which further empirical and theoretical study can build. Indeed, it is my hope that this volume may serve as a source of new ideas, and new paradigms for further study.

There are, of course, a number of contributions which bear directly on the reported research and should be mentioned. As far as I know the discussion of levels of theorizing is original although, as can be seen in the Epilogue, educational implications of the theory share certain things in common with what Simon (1969) calls *Sciences of the Artificial* (cf. Scandura, 1970a, 1972a). In addition to my own work, Chapter 2 obviously draws on a variety of mathematical notions. I have been influenced particularly by Nelson (1968), Suppes (See Volume II, Chapter 1), and Freyd (1964). The basic material in Chapter 3 is included in a variety of texts on logic and mathematical foundations. I found Stoll (1961), Robinson (1965), Nelson (1968), Lightstone (1964) and Rogers (1967) most helpful. Chapters 4, 5, and 6 are mostly original although they have gained much by analogy to formal linguistics, especially the writings of Chomsky (1957, 1963), and to logic, especially the material on natural deduction systems by Corcoran (Volume II, Chapter 6). Also see Prawitz (1965), Anderson and Johnstone (1962), and Kalish and Montague (1964). Chapters 7, 8, 9, and 10 were for the most part written independently although in retrospect they have many characteristics in common with the earlier work of Miller, Galanter and Pribram (1960) and Miller and Chomsky (1963). There are also certain relationships with recent work on adaptive control systems in learning, as represented in the work of Pask (e.g., 1966). One of my students, Francine Endicott, also came across Taylor's (1960) article, "An Information Processing Approach to Motivation," which anticipates Chapter 8 in part. Unfortunately, I was not aware of this paper at the time the chapter was written, as this might have saved me many hours of having to rediscover an important concept already alluded to by Taylor. As described in Chapter 10, my work on permanent memory bears important similarities as well as differences with theories proposed by others (cf. Norman, 1970). In addition, the discussion of information processing is generally compatible with certain views first discussed by Miller (1956). Finally, I would like to acknowledge a general intellectual debt to Robert M. Gagné, whose writings had so great an effect on my earlier thinking.
2. RELATION TO TRADITIONAL THEORIES

It may be instructive to begin our treatise by placing the to-be-proposed theory in the context of more traditional behavior theories. Specific attention is given to some of the major questions that must be asked of any viable theory of complex human learning.

The first question that must be considered is how to characterize the knowledge construct. Ignoring the many particular variants, there seem to be only two basically different ways of approaching this task—in terms of rules or in terms of associations. Even here recent theoretical evidence (See Volume II, Chapter 1; also see Suppes, 1969a, 1970; Arbib, 1969; Scandura, 1970b, 1970c) shows a close relationship between the two so that to some extent even this distinction may eventually disappear. The approach described below, for example, is fully operational (behavioral), which is a primary requisite of neoassociationistic formulations, but at the same time, the basic knowledge construct is taken to be the rule.

In line with the finitary nature of the mind, knowledge involves at most a finite number of rules. The major contribution of this volume is that of making explicit the laws which govern the interactions among the characterizing rules. In particular, rules which act on other rules are specifically allowed—at any level (i.e., rules may act on rules which act on ..., etc.). Most present views of competence are more restrictive. As we show in Chapters 4-5, even so-called transformational grammars are just a special case of higher-order rules—higher order rules which act on degenerate rules. The only (non-degenerate) higher order rule in traditional competence theories is that of rule composition. But, in such theories composition is viewed as a way of showing how certain input-output pairs may be generated by composing rules (i.e., applying one after another) and is not part of the rule set itself.

A second question to be answered concerns the ontogeny of knowledge. Is the human organism a blank slate on his arrival into the world, or does he come "wired in" with certain innate competencies? Few serious investigators, of course, still believe in a sharp dichotomy along these lines, but there is serious question as to the limits imposed by innate competence. Does innate competence really determine the ability to learn a language, for example, or is environment the crucial variable? What about abstract intelligence? What is the fundamental nature of maturation? These are just some of the fundamental questions that must be asked concerning the growth of knowledge.

The present view is that human beings are born with innate competencies
and, further, that these "instincts" may be characterized as rules. Maturation, then, is viewed as a physiologically-based process by which new innate competencies, also rules, become available. The critical job of the behavior theorist in each case is not to explain how these competencies come about but more simply to just identify what they are. His main task, of course, is to show how built in competence interacts with the environment to produce new knowledge. Only the latter task is explicitly treated in the present volume. Some attention is given to the former in Volume II, Chapter 7.

The third question involves the relationship between knowledge (competence) and performance. Here, there are two major subquestions. What type of theory—conditioning or information processing? And, if the latter, are rules to be attributed to the knower or are they simply introduced by the theorist to account for observed behavior? The first subquestion is easy. The to-be-proposed theory is clearly of the information processing variety; learning is the result of internal operations and not of contingencies between overt stimuli and responses. Information processing theories generally view learning as a problem solving process and mine is no exception. The key difference basically hinges on the precise specification of an intrinsically simple mechanism which accounts for simple performance as well as learning (and problem solving). Even motivation is governed by this mechanism, thereby giving more precise realization to Taylor's (1960) suggestions concerning an information processing approach to the subject.

In regard to the second subquestion, we stand on both sides of the fence. In agreement with Bourne, Ekstrand, and Dominowski (1971) and others of the "rule-following" persuasion, my view is that rules are used to account for behavior—but, in the specific sense that rules are introduced by the observer qua competence theorist to account for classes of stimulus-response pairs which are of interest to him and which he identifies prior to making any observations (of actual behavior). I also agree with Piaget and others who feel that subjects actually do use rules to generate behavior. Indeed, the learning and performance mechanisms introduced in Chapters 7 and 9 are assumed to operate on the rules known to the behaving subject, not those introduced by the competence theorist. As pointed out above, of course, in order for a theory to be operational it must provide some means of assessing behavior potential. Furthermore, what a subject knows must be "measured" relative to the rules introduced by the competence theorist. Because of the close relationships involved, predictions

---

1I believe that there is a strong analogy between "rule" as used here and "ruler" as used, say, in elementary physics. Both serve (continued)
concerning performance will be in error just to the extent that the under-
lying theory of competence is incompatible. This fact suggests that in
addition to the traditional criteria of power and parsimony, competence
theories—if they are to accurately reflect human knowledge—must also
meet certain performance criteria (cf. Chapter 6).

The fourth question pertains to memory and the limited capacity of
humans to process information. Here, again we take an information proces-
sing view as opposed, for example, to the older and more empirically based
interference, decay, and consolidation theories of memory. In my view,
the theory proposed in Chapter 10 does not so much run counter to these
theories. Instead, the theory is predicated on the belief that decay and
interference themselves, for example, are not adequate bases for explana-
tion, but rather are things to be explained. In effect, the present view
is that a deeper level of explanation is required.

By way of conclusion, I would like to make clear one very important
way in which my theory differs from most present day information processing
theories. In the latter, it is meaningful to ask such questions as: How
do subjects remember \( X \)? How do subjects learn \( Y \)? Or, solve problem \( Z \)?
We implicitly assume that there is a unique answer to such questions.

In my theory, such a question would be meaningless. There are many
different possible ways in which any particular subject might perform in
such cases.

A major problem for the theory is that of determining the basis upon
which any particular subject will perform. Theoretically, of course, there
are any number of possibilities. What keeps the number of possibilities
within bounds in real life is the common culture shared by the observer-
theorist and subject. In line with my comment above, the measuring units
selected by the observer-theorist, because of this common culture, are likely
to be compatible with the rules actually known to the subject. It is this
fact which makes it possible to even consider the possibility of a deter-
ministic theory concerned with the behavior of individual subjects in
particular situations.

---

(continued) as a device for measuring things of interest. Units of
measurement are introduced as a way of measuring quantities while rules are
introduced as a way of measuring behavior.
In spite of the increasing attention being given to rule-like processes in psychological theorizing, there is no commonly accepted definition as to just what a rule is, or what kind of behavior knowing a rule makes possible. Different investigators, for example, talk about rules for combining attributes to generate a category (Haygood & Bourne, 1965), for selecting alternatives in discrimination learning (Levine, 1966), for performing mathematical operations (Scandura, 1966, 1968, 1969a, 1969b), for generating new rules (Scandura, 1970b, 1971a; Roughead & Scandura, 1968), and for generalizing or restricting given rules—i.e., replacing constants with variables or vice versa (Scandura & Durnin, 1968). Attention has also been given to sequential rules, which operate over time, and to executive or control rules (e.g., Reitman, 1965), which determine how other rules are to be used. Furthermore, it has been common practice to use the term "rule" when referring both to behavior and to theoretical constructs used to account for behavior.

Section 1 of this chapter is concerned with the general nature of rules and rule-governed behavior. In Section 2, deeper and more precise characterizations are introduced for both rule-governed behavior and rules which underlie such behavior. Section 3 shows how the behavior potential of individual subjects relative to given classes of rule-governed behavior may be represented formally and how such behavior may be accounted for.

1. RULES AND RULE-GOVERNED BEHAVIOR

1.1 Rule-Governed Behavior

The major thesis of this chapter (indeed, the totality of Part I) is that all human behavior is basically rule-governed. For the present, however, no commitment is made as to whether this statement should be
interpreted to mean that behavior is the result of subjects' actually using rules (to generate the behavior) or whether rules are just constructs introduced by observers to account for observed behavior. (This distinction is of major theoretical importance and is considered in Sections 1.2 and 3.) In this section, it is sufficient to simply assume that rule-governed behavior exists.

Our immediate task is to describe some of the kinds of rule-governed behavior which subjects typically exhibit. Perhaps the simplest type of rule-governed behavior involves associations. Thus, saying "MUR" every time "ZUG" appears can be attributed to an association between the nonsense syllables "ZUG" and "MUR." Rule-governed behavior may also be attributed to concepts. In this case, the behavior involves sorting a large number of distinct stimuli into two piles, exemplars and non-exemplars (of the concept). The same sort of classificatory behavior may be further generalized, of course, to include any finite number of different responses.

In general, the number of distinct responses may be unlimited (i.e., denumerably infinite). For example, in adding numbers, giving different numbers as responses to different pairs of numbers in no way diminishes the fact that the responses are all sums, which can be generated in very much the same way. The same, of course, may be said for subtraction, multiplication, division, or for that matter any other computational task. (Clearly, it would take little imagination to view taking square roots or, say, finding the greatest common divisor of two natural numbers, in the same way.)

Once learning has taken place, responding in a discrimination learning task may also involve an indeterminate number of different responses. Thus, a subject might learn to "choose the larger of two objects presented." Such tasks as substitution into algebraic expressions can also be accomplished in a strictly mechanical manner. In addition, rule-governed behavior may occur in solving equations and, indeed, in solving many different kinds of simple problems. A person who can solve one mixture problem, for example, can typically solve other mixture problems of the same general type. Even making simple logical inferences may be considered to be a rule-governed activity. Consider syllogistic reasoning. In this case, for example, we can automatically infer (respond) that "Socrates is mortal" on the basis of the compound stimulus, "All men are mortals," and "Socrates is a man." Exactly the same sort of inference can be made in any similar situation, of course, and this is where the rule-governed behavior comes in.

In each of the above examples, not only does the behavior (the responses) appear to be rule-governed but it is reasonably clear just what the subject
What is a Rule?

is responding to. That is, we can, in the language of behavioral psychology, say which response is being given to which stimulus. There are many examples of rule-governed behavior, however, where the effective stimuli are not so easily detectable. The following task, in which the subject is presented with a row of six empty boxes on each trial, provides a simple illustration. On the first trial, the subject puts an "X" through the first box; on the second, an "X" through the third box; and on the third, an "X" through the fifth box. On the fourth trial, the "X" goes in the first box, and the sequence is repeated. The behavior is clearly rule-governed, but the (nominal) stimuli on the various trials are all essentially identical. At first glance, it is equally unclear what the stimuli are in writing the number series: 1, 4, 9, 16, 25, 36, ... . Obviously, each successive term can be generated by a simple rule (squaring) but again the stimuli to which the subject is responding are neither immediately detectable or observable. Proving theorems in mathematics is basically similar. A given proof may differ from others as to detail but the same general pattern may be repeated over and over again in a large number of different situations, even in quite different fields. It is rarely clear, however, just what stimuli a theorem prover is responding to in any given situation.

In general, rule-governed behavior may be said to have been observed when it is possible to tell (in advance) what a subject's next response will be. In some cases, such as in adding numbers, the predicted responses may be quite different from those on which predictions are based. The observer must extrapolate. In other cases, such as in sorting exemplars and nonexemplars, one or more responses are simply repeated over and over again in sometimes different circumstances. In all cases, however, the responses subjects make provide the key in detecting rule-governed behavior. Overt stimuli play a distinctly subordinate role. As we just saw above, there are many situations in which it is not even clear what the stimuli are.

1.2 The Rule Construct

A basic problem for psychology is how to account for rule-governed behavior. For reasons which are given below, we assume that the underlying mechanisms may be viewed both as constructs introduced by observers to account for such behavior and as mechanisms which subjects actually use. Throughout this section we shall be concerned with rule-governed behavior which has actually been detected by an observer and, hence, any mechanism
introduced by the observer to account for the behavior could be used by
the subject to generate the behavior and vice versa.

Traditionally, there have been two basic approaches to this problem.
Some theorists have chosen to elaborate on or to extend the S-R mediation
language (e.g., Berlyne, 1965; Staats & Staats, 1963). Others have shame-
lessly preferred more cognitive, or rule-based, formulations (Bartlett,
approach is to be preferred is perhaps based as much on a philosophy of
science as on psychology per se. The former approach appeals more to those
who want their theories and basic formulations grounded directly in empiri-
cal data. They have a reasonably precise language now, one which relates
specifically to behavior, and do not want to give it up without good reason.

Accounting for rule-governed behavior in terms of associations, however,
can become extremely cumbersome (e.g., Scandura, 1967a, 1968). The associa-
tive networks required generally involve far more detail than is necessary
to account for the behaviors (responses) observed. Indeed, until Suppes
(1969a, 1969b; also see Arbib, 1969) showed that any finite connected autom-
on can be characterized in terms of a finite number of S-R associations,

A major thesis of this section is that rule-governed behavior can be
accounted for more simply and directly in terms of rules. (The terms
"schema," "principle," and so on are simply different names for the same
type of construct.) A rule can be denoted by a function whose domain is
a set of stimuli and whose range is a set of responses. The concept and
the association become special cases. A concept can be represented by a
function in which each stimulus is paired with a common response, while an
association can be viewed as a function whose defining set consists of a
single S-R pair.

This still does not say, however, just what a rule is. Thus, a func-
tion can be defined either as a set of ordered S-R pairs, in which there
is a unique R for each S, or as an ordered triple. The denotation of a rule
(i.e., class of S-R behaviors which can be generated by a rule) seems best
characterized by the former type of definition, but the rule construct
itself conforms more closely to the latter type of definition involving a
set of inputs, a set of outputs, and a connecting operation.
Consider, for example, the task of summing arithmetic series (e.g., 1 + 3 + 5 + 7 + 9). In this case, any one of an equivalence class\(^1\) of overt stimuli (like the sign, "1 + 3 + 5 + 7 + 9") may represent the same number series (i.e., 1 + 3 + 5 + 7 + 9). Each such equivalence class serves as an effective (functionally distinct) stimulus. Effective responses (sums) may similarly be thought of as equivalence classes of overt responses (e.g., "25"). The denotation of the rule, then, consists of the set of ordered pairs whose first elements are equivalence classes of representations of number series, and whose second elements are equivalence classes of representations of their respective sums.

Underlying rules, however, are constructs and probably more naturally thought of not as acting on effective stimuli (responses) themselves but on properties of the entities denoted by these effective stimuli. Thus, for example, the property of having "a common difference of two between adjacent terms" refers to the number series, 1 + 3 + 5, and not to its name, "1 + 3 + 5." Note that a distinction is being made between the entity (e.g., number series) and the equivalence class of representations of that entity. However, since there is a one-to-one relation between equivalence classes of overt stimuli (the signs) and the abstract entities denoted, we can ignore the distinction here. (This distinction is raised again in Chapters 3 and 5.) These properties, in turn, determine (via the rule) other properties (of the responses). One rule for summing arithmetic series, for example, may be represented by the expression \([(A + L)/ 2]N\) where A refers to the first term, L to the last term, and N to the number of terms of the arithmetic series in question. The critical inputs associated with this rule are triples of values of the dimensions A, L, and N (e.g., A = 1, L = 7, N = 4). These triples may be viewed as (composite) properties of the entities denoted by the stimuli. We may refer to these critical properties as response determining (D) properties. The set of outputs consists of response properties (numbers) derived from the properties in D. These properties (numbers) determine equivalence classes of number names—e.g., the number property, 16, which is the sum of the series,

---

\(^1\)By an equivalence class of overt stimuli (responses) or an effective stimulus is meant a class of overt stimuli, each of which has the same set of defining properties. The term "effective" is used to emphasize that we are talking about the stimuli and responses "effectively" operating in the situation rather than the overt stimuli and responses themselves. Thus, for example, the stimuli "5" and "five" would, for most purposes, count as the same effective stimulus since they both represent the same number. The stimuli "5" and "6," on the other hand, would correspond to different effective stimuli. In previous papers, Scandura (1966, 1967a) used the term "functionally distinct."
1 + 3 + 5 + 7, defines the equivalence class of all signs (e.g., "16," "8 + 8") which denote 16. Notice, however, that these number properties may also be viewed as properties of the series themselves. In this role, the number properties are called sums, which just happen to be properties of arithmetic series which can be derived from other presumably more easily determined properties, like the first term and the number of terms.

In effect, a rule may be defined as an ordered triple (D, O, R) where D refers to the determining properties of the stimuli, and O to the combining operation or transformation by which the derived properties (of the responses, R) are derived from the properties in D. The rule associated with the usual addition algorithm, for example, may be defined as follows: D = set of all pairs of whole numerals, R = set of all whole numerals (which may act as sums), and O = add the units digits, if the sum is greater than 9, carry one and ... (as in the usual addition algorithm). It is also a simple matter to construct rules which account for conceptual and rote (association-governed) behavior. For example, a rule for distinguishing between exemplars (e.g., cards with exactly five objects) and nonexemplars (cards with more or less than five objects) may be characterized:
D = \{0, 1, 2, 3, ...(i.e., numbers of objects)\}, O = Count the number of objects, if five, put in container A; otherwise, put in container B, R = \{Being in A, Being in B\}. Rote behavior is even simpler. The association, MUR \rightarrow ZUG, may be accounted for in terms of the rule (association):
D = \{MUR\}, O = say ZUG, R = \{ZUG\}. More ingenuity is involved in constructing rules for generating certain complex behaviors, such as in making logical inferences, but exactly the same principles are involved.

Actually, rules may be characterized uniquely in terms of the operation (O), together with the set of inputs in D. Together, the two provide a sufficient basis for generating the outputs in R.

In accounting for rule-governed behavior, therefore, it is sufficient to identify a set of input properties and an operation such that application of the latter to the former yields precisely the same pattern of output properties as was observed. For denotative purposes, the specific nature of the operation is immaterial; it matters only which response is paired with each stimulus. We shall have many occasions below where we shall want to refer to rule-governed behavior independently of actual observation. In grading test results, for example, the general paradigm involves comparing observed behaviors (called test responses) with other behaviors (called solutions) which are somehow given a priori. (The question of how to represent such relationships formally is considered in Section 3.)

There is obviously a close relationship between inputs and outputs,
What is a Rule?

on the one hand, and stimuli and responses, on the other. (In view of this relationship, rule denotations are referred to below as classes of rule-governed behavior. From now on when we speak of a class of rule-governed behaviors we are referring to a class of S-R pairs and not just the responses.) They are not the same, however. In adding numbers, for example, the traditional S-R view would be to identify stimuli with such expressions as "14 + 65." But, in fact, most addition rules act on only certain aspects of such stimuli. Specifically, the ordinary addition algorithm acts only on those properties of such expressions which define specific numerals. The "+" sign, for example, is extraneous.

If this distinction seems somewhat artificial and unimportant, again consider the task of summing arithmetic series of the form, 1 + 3 + 5 + ... + (2N -1). In this case, the overt stimuli would be such expressions as "1 + 3 + 5 + 7," "1 + 3 + 5 + 7 + 9 + 11," and so on. The associated responses (sums) can be generated by the rule denoted N^2, where N refers to the number of terms. Clearly, the squaring operation does not operate on the given expressions themselves, but rather it operates on a particular property of such expressions (the number of terms). The rule can be characterized: D = the set of numbers, 1, 2, 3, ..., representing the number of terms in a series, 0 = the squaring operation, and R = the set of all perfect squares (1, 4, 9, 16, 25, ...).

This distinction between overt stimuli and responses, on the one hand, and input and output properties, on the other, may seem to be a relatively minor point, but it is not. It represents a major departure from the kind of thinking associated with S-R neoassociationistic psychology. In S-R psychology, a major tenet is that predicting behavior is dependent only on the present stimulus and the past history of stimulation to which a particular subject has been exposed. Specifically, the present view rejects the idea that overt (and/or potentially overt) stimuli cause behavior. We assume instead that behavior is caused by rules, an underlying construct—in effect, that subjects actually do use rules. Stimuli simply provide the occasion for responding. (In Chapter 7, we shall see that the rules used on particular occasions depend on subjects' goals.)

In this cognitive based behavioral approach, the emphasis is on responses. Instead of concentrating solely on stimulus manipulation, a major task in explaining and/or predicting (new) behavior is to construct a rule which accounts for the observed behavior. In order to accomplish this, of course, the investigator must first identify just what the critical aspects of the behavior are, that is, what response properties must be attended to and which may be ignored. In writing sums of addition problems
for example, it is quite immaterial whether the numerals are, say, written in pencil or ink, large or small, and so on. The only properties which matter are those which serve to uniquely specify the numerals. In effect, the properties attended to by an observer define *equivalence classes of overt responses*. Association-governed behavior, for example, involves just one response equivalence class. Thus, when we speak of a response like the nonsense syllable, RES, what we really mean is an equivalence class of overt responses, each of which has all of the essential properties of RES. "RES," "Res," "res," or even "RES"(trial 1), "RES"(trial 2), ..., for example, are all equivalent in this sense. In general, of course, rule-governed behavior may involve any number of response equivalence classes. For example, in observing performance on a series of addition tasks, an adult observer is almost certain to distinguish between numerals which represent different numbers. Thus, numerals like "6," "six," and "6," would be placed in one equivalence class, while the numerals, "8," "eight," and "8" would be placed in another. The important point of all this is that the properties the observer attends to determine what the effective (behaviorally distinct) responses are.

Introducing a rule to account for a class of (effective) responses amounts to specifying both a class of inputs (properties) and an operator from which these responses may be generated. These critical stimulus properties, in turn, effectively define equivalence classes of overt stimuli. In the arithmetic series example above, for example, each specific value of N (a property) in 1 + 3 + 5 + ... + (2N - 1) defines an equivalence class of that form. For example, N = 3 defines the class {1 + 3 + 5, 1 + 3 + 5}, one plus three plus five, ...}. The same is true of even simple behavior, such as that involved in repeating a given nonsense syllable (e.g., RES). Introducing an association to account for the behavior, then, amounts essentially just to specifying the stimulus properties to which that response is to be given. (The operations in associations are all just simple one-to-one connections.) Thus, for example, if an investigator observes that RES is given whenever "STI," "STU," or "STO" are presented, he might be inclined to equate the input class, D, with the properties "S" in the first position, "T" in the second, and any vowel in the third. This defines the stimulus equivalence class, {"STI," "STU," ...}

It is important to notice that the critical stimulus properties in the domain of a rule define effective stimuli only insofar as these properties specify values in given and generally larger classes of stimuli. Thus, for example, the number of terms effectively partitions the class of expressions of the form 1 + 3 + ... + (2N - 1). Even the stimulus
properties in the domain of a simple association are usually (and implicitly) interpreted with a restricted class of stimulus expressions in mind (e.g., the stimuli may not contain more than one word, or sentence). Such classes of stimuli constitute the stimulus domains of particular rules.

Similarly, in introducing a rule to account for observed rule-governed behavior, the observer must necessarily key on certain aspects of the overt stimuli (where they are present). Whether the stimuli are observable or not, however, the rules introduced do not deal with the encoding process. In the present case, where we assume that subjects actually use rules, introducing such a rule amounts essentially to making certain assumptions about the encoding and decoding capabilities of the subject(s) in question. (In order to use a given rule, a subject must be capable of the encoding and decoding required.) When in doubt, of course, the observer may choose to assume less about the subject by absorbing into the operational aspect of the rule as much of the encoding and decoding as he wishes. For example, instead of just assuming that a young child can write numerals, the observer may actually spell out smaller-sized steps by which such numerals may be constructed (e.g., draw a vertical line, then at the top draw...). (In sections 2.2 and 2.3, we shall see how encoding and decoding (as well as the operational aspects of a rule) may be broken down into units as small as one might find necessary to reflect a subject's actual capabilities.) No matter how detailed the account, however, there will always be encoding and decoding gaps.²

These gaps are at once the major strength and the major weakness of the present approach. They are a strength in that they make it possible to deal with arbitrarily complex behavior (by ignoring unessential detail). They are a weakness in that they introduce the need for a complementary theory of perception (cf. Chapters 5 and 9).

²Nonetheless, by allowing certain observables to act as primitives, encoding and decoding rules can make direct contact with reality (i.e., stimuli and responses). The basic idea is that any observable may be thought of as composed of certain indivisible elements. The strings, AB, ABA, BBBAB, ..., for example, may be constructed from the primitives A and B. Furthermore, properties of these strings can be thought of as generated by application of rules which act on these primitives. Thus, for example, picking the "first term" of a string becomes a meaningful operation since encoding rules may operate directly on the primitives rather than on other properties of the strings (stimuli). Similarly, the number of terms in a string can be generated by actually constructing a numeral to represent this number. In the case of a "tally mark," for example, this could be done by simply constructing one slash for each primitive in a given string.

This process directly parallels that used in defining formal systems (cf. Chapters 3 and 5) and serves as a basis for much of the current work in artificial intelligence on perception.
Notice that introducing rules to account for observed behavior bypasses questions relating to the encoding and decoding, respectively, of overt stimuli and responses. Although they may be thought of as insertion into and extraction from equivalence classes (for details see Chapter 5), respectively, these processes can also be viewed in terms of environmental (observational) theory. Their nature is determined as much by what the observer attends to as by what the subject does. Unattended to aspects of response situations may simply be ignored for purposes of theorizing about the subject's behavior.

Defining stimuli in terms of constructs, rather than the reverse, has the major advantage of putting what S-R neoassociationists call respondent and operant behavior on the same plane. In particular, identifying a rule which accounts for observed rule-governed behavior has the effect of specifying a class of effective stimuli, irrespective of whether these effective stimuli have direct overt counterparts or not. To see this, recall the example of rule-governed behavior above in which X's were placed in various boxes of a six-box array, in a definite sequence. This behavior can be generated without error by the following rule: "If the previous X was in the fifth box, put the X in the first box. Otherwise, put the X two boxes to the right of the previous X." Assuming that the first X is given, this rule makes it possible to generate the indicated behavior without error.

The critical stimulus properties in the domain of this rule are precisely those that define the array of six boxes, together with those properties specifying where the X was placed on the previous trial. At the time a given response is made, the former properties are indeed observable, but the latter are not. In the words of a behaviorist, the response is elicited at least partly by a "stimulus trace."

Allowing rules to specify effective stimuli, then, has the effect of eliminating what could be an embarrassing question for neoassociationists. When is a prior effect to be considered as part of the eliciting stimulus, and when is it not to be so considered? What, for example, are the effective stimuli associated with the rule-governed behavior involved in extending the sequence 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... ? This happens to be the well-known Fibonacci series in which each term (except the first two) is the sum of the preceding two terms. Surely, in more complex behavior, the "controlling stimulus" event may be even more difficult to determine. Indeed, the effective stimuli may have no overt counterparts whatever (i.e., the stimuli may be entirely internal) or the corresponding overt stimuli may have occurred so long before as to transcend the investigator's observational capacity.
By giving primary attention to the notion of stimulus, the neo-associationist has been forced into an untenable position. True, he can specify his constructs (associations) in terms of the stimuli where they are observable (as with respondent behavior). But where they are not, he must take a completely different course—that is, devise an independent theory of operant behavior. Specifying a rule which accounts for observed behavior has the effect of reducing this dichotomy to a difference in terminology (to be sure, a difference with certain practical implications). (This says nothing of the fact that introducing rules makes it possible to account for behaviors far more complex than anything attempted by neo-associationists. Even proofs of mathematical theorems, for example, may be accounted for in terms of rules (see Scandura, 1971a; also Chapter 6).)

In general, the same rule-governed behavior may be accounted for in any number of ways. To be more exact, if even one rule can be found for generating a given class of behaviors, then there are a denumerably infinite number of ways for generating precisely the same behavior. Practically speaking, of course, there are relatively few such procedures that are likely to be used by individual subjects (to generate the behavior) or to be introduced by most observers (to account for the behavior). Giving sums in simple addition problems, for example, might reasonably be accounted for by: (a) some version of the ordinary addition algorithm, (b) incrementing (by one) the first number as many times as the second, (c) selecting two arbitrary disjoint sets corresponding to the two given numbers (e.g., two fingers on one hand and four on the other) and then counting the total number of elements (in the union). Although all of these rules account for the same rule-governed behavior, the ability to use one of them does not necessarily imply the ability to use any of the others. Thus, a young child may be able to find the sum of two (small) numbers by counting but have no idea whatsoever that such a thing as the addition algorithm even exists.

The number of feasible ways of generating responses to stimulus nonsense syllables can be especially large. Instead of utilizing all of the defining properties of the stimulus "sti," for example, a subject might be just as apt to key on the first letter, consider the overall shape of the syllable, embed the syllable in a familiar word (e.g., stimulus, stimulation), and so on. Each such association would carry with it different implications for behavior, particularly on transfer tasks. Thus, for example, one subject might embed the (overt) stimulus, "Sti," in the word, "Stimulus." In this case, the same response would be expected to overt stimuli like, "Stim," "Stimul," or "St." Another subject might generate
the response by keying only on the first letter of "Sti," requiring only that the first letter be "S" and that the stimulus expression contain three letters. In this case, one would expect the response to any three-letter stimulus beginning with "S," including such expressions as "Sux" and "Stc" as well as "Sti." It is just this sort of arbitrariness, in my opinion, which makes paired-associate and serial learning so difficult to study. Contrary to prevailing opinion, the study of learning with highly structured meaningful material may actually be easier.

It is also my conviction that failure to be aware of differences in the way rule-governed behavior may be generated may also lead to unnecessary confusion in experimental research on complex learning. Consider, for example, the important problem of number conservation. Several decades ago, Piaget observed that some young children, between the ages of five and seven, were able to compare the number of objects in two collections correctly when they were arranged in some ways but not in others. For example, even the young child will say that two collections contain the same number of objects when the respective elements are physically paired in a one-to-one fashion. If, however, the objects in one collection are spread out so that they appear to cover a greater area, the nonconserving child typically will say that this collection contains more objects. Piaget has maintained that conservation of number develops gradually and tends to appear spontaneously with time, due to a wide range of experiences, and is not subject to specialized training.

During the past few years, any number of investigators have attempted to prove Piaget wrong. In a recent study, for example, Wallach & Sprott (1964) were apparently able to make conservers out of nonconservers during the course of a short-term experiment. They trained subjects, through standard reinforcement procedures, to respond correctly whenever one of the collections was transformed in one of several ways. The transformations included rearranging the objects, removing objects, and/or adding objects. The experimenters were, indeed, successful in training their subjects on this task. Furthermore, the subjects were able to transfer their new-found ability to other conservation tasks in which the materials were changed.

Do the data of this experiment contradict Piaget's contention that conservation of number is not subject to (short-term) experimental intervention? On the surface, this would appear to be the case. A more detailed look at the experimental task, however, indicates that the experimenters probably failed to include a crucial test of conservation. In particular, number conservation corresponds precisely to the mathematical notion of
What is a Rule?

cardinal number. Each cardinal number (e.g., 1, 2, 3, ...) may be defined as the class of all sets which can be put into one-to-one correspondence with one another. A child who has learned to conserve (cardinal) numbers, then, has learned a rule for determining whether or not it is possible to pair in a one-to-one fashion the objects in two (or more) arbitrary collections.

The subjects in the Wallach and Sprott study easily could have learned to give the correct responses without such a rule. That is, it is quite possible that they could have learned a simpler rule, which nonetheless still made it possible to respond correctly to the tasks presented. All they had to do was remember the previous response (yes or no) and observe the sort of transformation the experimenter made. If the previous response was "yes" and a rearrangement was performed, the appropriate response would still be "yes"; the same would be true for "no". If the experimenter added to or subtracted from one of the collections, the appropriate response could be determined with equal ease. (Note the similarity between this rule and that involved in the "six boxes" example above. In both cases, making a correct response depends on remembering the previously correct response.)

Contrast the Wallach and Sprott tasks with one the investigators did not consider—that of adding to or subtracting from one of the collections without the subject's knowledge, together with rearrangement where necessary to avoid correct responding on the basis of direct perception. Unlike the experimental tasks, a subject would be unable to give the correct responses without going through the process of one-to-one pairing in one form or another. Clearly, the authors would not have made the claim they did had they first identified the kind of rule(s) number conservation involves. Because of this, the sets of stimuli and responses they identified and used during their testing were inadequate to unambiguously determine conservation of number.

Before moving on, one final word of caution is in order. In dealing with meaningful behavior, it often is useful to distinguish between rules which operate on (equivalence classes of) signs used to denote meanings, and rules which operate directly on the meanings themselves. As it is usually interpreted, for example, the ordinary addition algorithm acts directly on signs. The rule is essentially one of symbol (digit) manipulation which can be carried out with no reference to underlying meanings (e.g., the symbols 5 and 6 in +76 give 11; the 11 gives 1 and carry 1; carry 1, 8, and 7 give 16). In this case, the elemental signs are treated as indivisible wholes and correspond to what automata theorists call
letters of an alphabet. The digit (numeral) 5, for example, is defined in terms of the set of all properties essential to being (the numeral) 5. None of the defining properties plays any special role. The corresponding equivalence class of overt stimuli includes such signs as "5," "5," "5−," and so on.

Adding numbers by counting, on the other hand, involves direct reference to underlying meaning. That is, the numerals to be added must be interpreted as (disjoint) sets. Counting the total number of elements, then, depends only on the property that the entities being counted are elements of the sets. Those properties which determine the specific nature of the elements are irrelevant. In general, rules which operate on meanings typically involve reference to a small finite number of properties. These critical properties define the equivalence classes of stimuli. (As we shall see in Chapters 3 and 4, this has much in common with axiomatic mathematics, where families of systems are specified by a finite number of properties called axioms. Other properties, called theorems, are generated from the axioms via application of rules of inference.)

Finally, we note that there is no uniform type of relationship between meaning properties and sign properties (which may be involved in corresponding rules.) Relationships may range from highly similar to having essentially nothing in common. Consider, for example, number series of the form $1 + 3 + 5 + \ldots + (2N - 1)$. In this case, the symbol "1," which represents the first term property, has little to do with its meaning. The number of terms (property) of a series of this form, on the other hand, corresponds directly to the number of numerals (property) of the sign used to represent the series. (By definition, signs which have given properties in common with the entities they denote are called icons (relative to those properties). Symbols, on the other hand, denote arbitrarily. The general question of reference is an extremely important one and the reader is referred to Chapter 7 for a more complete discussion.)

2. MORE PRECISE SPECIFICATION

2.1 Characterization of Rule-Governed Behavior

The assumption that rules can account for all human behavior raises the question of precisely what kind of behavior subjects are capable of. Since all rules generate functions, one might suspect that human beings are capable of generating any class of behaviors (i.e., class of effective S-R pairs) which is a function—or, equivalently, that every function can be generated by a rule.
What is a Rule? 23

This is not quite true, however. While rules exist for generating many kinds of functions, there are functions for which no one has yet found a generating rule. Indeed, there are functions for which no such rule even exists. (Nonetheless, it is always possible to devise a rule (e.g., an "association") which works in any particular S-R instance of the class. In general, devising such a rule belongs to the domain of problem solving. See Chapter 7.)

The central problem is to identify which kinds of functions can be generated by rules and which cannot. Fortunately, the answer to this question is generally agreed upon by logicians and others working in recursive function theory. According to what is known as Church's Thesis (Rogers, 1967, 20-21), a partial function can be generated by algorithm if and only if the partial function is a partial recursive function (or more accurately, a recursive partial function). Partial recursive functions have been characterized in a number of different ways, but fortunately, the major formulations of Church, Post, Kleene, Markov, and Turing have all been shown to be equivalent.

This is not the place to dwell on such issues as they have been treated in depth by many others (e.g., Nelson, 1968; Rogers, 1967). Nonetheless, it may be instructive to consider the notion briefly as it might relate to behavior. Generalizing from Rogers (1967) and Nelson (1968), the class of partial recursive functions is the smallest class of functions such that:

1. All constant valued functions are in the class. In terms of behavior, constant functions correspond to generating common (arbitrary) responses to one or more stimuli.

2. All successor functions are in the class. In more general behavioral terms, successor functions correspond to functions which "do the same thing" to each stimulus element in their domains (e.g., increment by one, make a quarter turn to the right, etc.).

3. All identity functions are in the class. These are functions which map one or more stimulus properties into themselves—as in mapping the triple (5, 7, 2) into 7, the value of its second element.

4. The compositions of any number of functions in the class are also in the class (e.g., if $f_1$ and $f_2$ are in the class, then $g = f_1 \cdot f_2$ is also in the class). In behavioral terms, composite functions may be thought of as first performing according to one rule-governed class and then another, where the outputs of the first class serves as the inputs of the second.
(5) If $h$ is a function of $k + 1$ variables in the class, and $g$ is a function of $k - 1$ variables in the class, then the unique function $f$ of $k$ variables satisfying

$$
f(0, x_2, \ldots, x_k) = g(x_2, \ldots, x_k)
$$

$$
f(y + 1, x_2, \ldots, x_k) = h(y, f(y, x_2, \ldots, x_k), x_2, \ldots, x_k)
$$

is also in the class.

(6) If $f$ is a function of $n + 1$ variables, the $n$-ary function $h$ whose values are

$$
h(x_1, \ldots, x_n) = \min y \ [f(y, x_1, \ldots, x_n) = 0]
$$

is also in the class. That is, the values of $h$ are determined by choosing the "minimum" value of $y$ which satisfies the equation

$$
f(y, x_1, \ldots, x_n) = 0.
$$

It is relatively easy to envision functions that can be constructed solely from functions of the first four types. For the most part, their interpretation must be left to the reader. Each such function can be thought of as a simple composite of functions of the first three types. Let us call such functions simple. "Make a quarter turn to the right, take three giant steps forward, and lie down and roll over," describes the operations involved in generating one simple function.

The fifth way of generating functions is somewhat different. Together with the first four ways, the fifth defines the class of primitive recursive functions. Intuitively speaking, these are functions in which different operations are performed on different elements in the domain. In particular, primitive recursive functions are equivalent to classes of functions in which the functions (in the class) are orderable and each can be generated from its predecessor by a common operation. The function $f(n, m) = n + m$ where $n, m$ are natural numbers, is one such function. It does exactly the same thing as does the class $\{g_0(n) = f(n, 0) = n; g_1(n) = f(n, 1) = n + 1; g_2(n) = f(n, 2) = n + 2; \ldots; g_j(n) = f(n, j) = n + j; \ldots\}$ where the function $g_j(n) = f(n, j)$ (for each $j$) corresponds to the rule "store $n$ and increment it $j$ times." In this case, each function $g_j$ (for $j \neq 0$) can be generated from its immediate predecessor $g_{j-1}$ by a function $h'$ which takes $g_{j-1}$ into the composite function

$$
h'(g_{j-1}) = s \cdot g_{j-1} = g_j
$$

where $s$ is the successor function generated by the rule "increment by one." In an important sense, then, any primitive recursive function may be thought of as generated from a given function by repeated application of another, first to the given function and then to the successive outputs (functions).

The relationship between this interpretation and the defining equations of (5) is camouflaged because of the traditional procedure in recursion theory of defining the $(y + 1)^{th}$ value of a function $f(y + 1,$
What is a Rule? 25

\[ x_2, \ldots, x_k \] in terms of its \( y \)th value \( f(y, x_2, \ldots, x_k) \). To see how the
\[ \text{equation } g_j = h'(g_{j-1}) \text{ relates to (5), we first rewrite the equation in terms of function values, giving} \]
\[ g_j(i) = h'(g_{j-1}(i)) \]
Substituting \( f(i,j) \) and \( f(i,j - 1) \) for \( g_j(i) \) and \( g_{j-1}(i) \), respectively, gives
\[ f(i,j) = h'(f(i,j - 1)) \]
\[ h'(f(i, j - 1)), \text{ in turn, can be rewritten as } h(i, f(i, j - 1), i - 1) \text{ as} \]
required in the original statement of (5). Note: In \( h'(g_{j-1}(i)) \), \( h' \)
is applied to \( g_{j-1} \) and the output \( g_j \) is applied to \( i \).
\[ h'(f(i, j-1)) \text{ is similarly interpreted.} \]

As it stands, (6) represents a rather special way of generating a
specific numerical function of \( n \) variables from a function of \( n + 1 \)
variables. For purposes of analyzing behavior it may be more useful to
reinterpret the procedure more generally. That is, procedure (6) might be
thought of as generating a class of functions of \( n \) variables from a given
function in \( n + 1 \) variables by substituting specific values for one of
the variables. Given the function \( \left( \frac{A + L}{2} \right)N \), for example, it is possible to
generate the class of functions \( \{ \left( \frac{1 + L}{2} \right)N, \left( \frac{2 + L}{2} \right)N, \left( \frac{3 + L}{2} \right)N, \ldots \} \).
The mechanism which generates the functions in this class may also be
viewed as a function in which the inputs are pairs consisting of \( \left( \frac{A + L}{2} \right)N \)
and a specific value \( (1, 2, 3, \ldots) \) to be substituted for \( A \). The rule
corresponding to this function has exactly the same effect on all of the
inputs; the same operations are involved.

This way of generating new functions is in some sense almost the
precise opposite of procedure (5). Primitive recursive functions
(requirements 1-5), recall, are viewed as ordered classes of simple
functions (requirements 1-4) in which each successive simple function can
be generated from the previous one by a single simple function. Just how
deep this relationship may be I cannot say but it does seem sufficiently
natural to warrant some investigation.

Although it has been convenient to mention rules, in discussing par-
tial recursive functions, it is important not to confuse the two. In
traditional logical terminology, the study of partial recursive functions
is said to be extensional. The study of mechanical procedures (rules) for
generating such functions is said to be intensional. In the chapters which
follow, the terms function or class of rule-governed behaviors (i.e.,
class of S-R pairs) will be used when the emphasis is extensional; rule,
when the emphasis is intensional.

For the reader familiar with automata theory it may also be worth
noting that recursively enumerable sets correspond to classes of rule-
governed behavior in which the stimuli are internal (or at least not obviously external). The elements in the recursively enumerable sets correspond to potentially observable responses, but the corresponding stimulus domains are unspecified. (This has the effect, of course, of providing more latitude in coming up with a rule to account for the behavior.)

Sets are said to be *recursive*, on the other hand, if there is an effective procedure (rule) for deciding whether any element in some given universe is in the set or not. As such, recursive sets correspond to conceptual behavior (i.e., classes of rule-governed behavior which can be generated by a restricted kind of rule called a concept). The elements in a *recursive set* correspond to concept exemplars, while those not in the set correspond to nonexemplars.

### 2.2 Characterization of the Rule Mechanism

As mentioned previously, the (D, 0, R) characterization tells only how the critical response properties are derived from the critical stimulus properties. It says nothing about how stimulus properties are determined (i.e., encoded) nor how response properties are converted into overt responses (i.e., decoded). It is implicitly assumed that the subject is able to accomplish the necessary encoding and decoding and, further, that all of the encoding is accomplished essentially simultaneously at the beginning, and all of the decoding, at the end. In addition, the operator in the (D, 0, R) characterization is treated as an indivisible whole. No attention is given to the various steps of which the operator may be composed.

While useful for many purposes, the (D, 0, R) characterization is inadequate where detailed analysis is required. Specifically, this characterization does not distinguish precisely between procedures having essentially the same operators and identical sets of determining and response properties, but where the order in which these properties are used and/or the degree to which the operations are performed internally may differ greatly. As long as it remains necessary to resort to ordinary English (or some other natural language) in describing such detail, there will probably continue to be points of ambiguity in research on structural learning.

In the following discussion, we shall show how such detail can be represented in a precise, unambiguous manner. Below, we give a *rationale* for the notions introduced, and show how this reformulation makes it
possible to distinguish between different rules having the same \((D, 0, R)\) characterization. Then, in Section 2.3, we show how rules can be represented schematically in terms of flow diagrams and directed graphs at any desired level of detail. Relationships between these schematic representations and associative networks, finite automata, and TOTE hierarchies are also described.

**Rationale:** Essentially, rules are nothing more than algorithms (mechanical procedures) for generating classes of responses from corresponding classes of stimuli. Since computers can be programmed to follow any algorithm, the detail involved in rules could be represented in terms of computer programs. To do so, however, might well introduce possibly misleading irrelevancies into the situation insofar as human behavior is concerned. Programming languages (including the powerful list processing languages frequently used in simulation studies) have been constructed for use with computers, not human beings. (As we shall see below, even flow diagrams have characteristics which do not necessarily parallel fundamental psychological processes.) Such characteristics have tended to make it difficult for psychologists to truly evaluate their worth and may, in fact, be misleading to the computer simulation specialists themselves by channeling their thinking about human behavior toward unproductive ends. As an example of such a characteristic, consider the notion of branching. As usually displayed, choice points are represented in a computer program as a sequence of binary choices. Human beings frequently choose from larger classes of alternatives. Similar characteristics in programming have been the emphasis on numerical computations and the tendency to insert all initial states into memory at the beginning of a computation. In short, the basic instructions in computer languages, as well as the subroutines derived from them, are defined as they are for the *convenience* of programmers, not to parallel basic human processes.

This does not imply that programming languages are intrinsically unable to handle the same kinds of problems that humans can. They can. The point is that the algorithms constructed, using such languages, typically have many characteristics which have nothing to do with human behavior. Programming languages are closely tied to computer mechanisms and, thus, may tend to obscure basic human phenomena.

For present purposes, it seems better to extrapolate the more general notion of a "program" in the abstract sense of automata and recursion theory (e.g., Nelson, 1968; Rogers, 1967) and to introduce modifications which seem desirable from a psychological point of view. Accordingly, the operational aspects \((0)\) of a rule may be decomposed into basically just
two kinds of (sub)rule: operation rules, which map properties into properties, and decision making capabilities or decision rules, which map output properties into a finite number of possible "next" rules. Intuitively speaking, an operator may be thought of as a finite directed graph (cf. Section 2.3) whose arrows are operations and whose nodes are decision making capabilities. (From now on, the term "rule" is used where (D, O, R) is considered as a whole; "procedure" refers to algorithms which have been decomposed into parts.)

The operator involved in the procedure $[(A + L)/2]^N$ for summing arithmetic series (e.g., $2 + 6 + 10 + 14 + 18 + 22$) provides a particularly simple example.

1. (a) Add A and L (e.g., $2 + 22$),
   (b) Divide the sum $(A + L)$ by 2 (e.g., $24/2$),
   (c) Multiply the quotient $(A + L)/2$ by N (and STOP)
   (e.g., $12 \times 6 = 72$).

Notice that in this case all of the rules are operational. Adding branching rules would be redundant, since the output of each constituent (operational) rule leads unequivocally to the next rule in the sequence (e.g., consider (1a) and (1b)). If we wished, however, we could insert immediately after rules (1a) and (1b) the branching rule,

   (d) If the output is a number (which it always is if the previous operator has been appropriately carried out), apply the next rule in the sequence; (otherwise STOP).

This procedure has the property that no matter which stimulus input (number series) it is applied to, exactly the same number of steps is involved. This is not true of all procedures. In general, the number of steps required may vary greatly according to the particular input in question. While the number of steps is finite in each case, there is no upper bound on the number of steps that might be required. Nonetheless, each procedure is composed of at most a finite number of different operation and decision rules, each of which may (or may not) be repeated during the course of an application. Such repetition is made possible by branching rules. Branching rules test each output of the immediately preceding operation rule against one or more criteria, and depending on which of the several possible criteria is satisfied, control is shifted to one or the other of the operation rules in the list (possibly the same rule).

Probably the most familiar test criteria involve equality. For example, does $X = 7$? Or not? There are, however, any number of other types

---

3 In my formalization of the theory (cf. Chapters 5 and 9) decision making capabilities have been defined as partitions.
What is a Rule? 29

used by human beings, even in doing mathematics. Thus, for example, one might also ask: (a) Is $X < 7$? $X > 7$? $X = 7$? (b) Is $X$ in the neighborhood of $P_0$? Or not? (c) Is $X$ a number? Or not? (d) Is $E$ an identity element for $X$ (i.e., is $E \circ X = X$ for all $X$)? An inverse element? Neither?

As an example of how branching rules operate, consider the procedure of adding pairs of numbers by incrementing (by one) the first number, $X$, as many times as the second number, $Y$. This procedure can be characterized by the following sequence of rules.

(2) (a) If $Y = 0$, apply Rule (b); otherwise, apply Rule (c).
    (b) Retain $X$ as the sum, and STOP.
    (c) Increment $X$ by 1, and reduce $Y$ by 1.

To interpret this procedure correctly, it is important to note that $X$ and $Y$ refer to information held (as we shall see in Chapter 10) in short-term store. This information may be changed during the course of a computation. Suppose, for example, that the number 3 is stored in $X$, and 2 in $Y$. Then each application of the incrementing rule (2c) increases the contents of $X$ by 1, and at the same time decreases the contents of $Y$ by 1. More specifically, the chain of events would go as follows: (2a) $Y \neq 0$, so (2c) is applied next. Rule (2c), then, increments the 3 in $X$ by 1, giving 4, and reduces the 2 in $Y$ by 1, giving 1. (Since (2c) is listed last, control is understood to go next to (2a).) Since the number (1) now in location $Y$ is still not zero, (2a) refers the computation again to (2c), which increments the 4 in $X$ by 1, giving 5, and reduces the 1 in $Y$ by 1, giving 0. This time, $Y = 0$, so control shifts to (2b)—retain the 5 in $X$ as the sum, and STOP.

It is important to notice that operating and branching rules deal only with internal information processing. Nothing has been said about how external information is internalized (i.e., encoded) so that it can be processed, or how internal information is made observable (decoded). In the previous illustration, for example, it was assumed that the numbers 3 (in $X$) and 2 (in $Y$) were available (internally) to the information processor from the start. Similarly, the computation was stopped with 5 stored in $X$; nothing was said about making 5 observable.

As indicated above, it has been implicitly assumed in most studies of rule learning, based on the (D, 0, R) construct (cf. Vol. II, Ch. 2), that all of the encoding is accomplished initially, and all of the decoding, after the computation has been completed. Clearly, this is not always the case. In carrying out the ordinary addition algorithm, for example, people typically deal in order with the units digits, the tens digits, the hundreds digits, and so on, during the course of computation. They also
write down partial sums, and often indicate overtly the numbers being carried. This has the effect of greatly reducing the amount of information which must be processed internally at any one time. (The importance of this in human behavior is referred to below, and dealt with in detail in Chapter 10).

In order to represent such detail, encoding and decoding rules must be introduced and interspersed at appropriate places in the sequence of rules used to characterize given procedures. Procedure (2), for example, might be revised as follows:

(3) (a') Encode the defining properties of stimulus "n + m" (i.e., put n and m in X and Y, respectively).
(b) If Y = 0, apply Rule (b); otherwise apply Rule (c).
(c') Decode the contents of X, and STOP.
(c) Increment X by 1, and reduce Y by 1 (go to a 1).

In this case, Rule (3a') effectively encodes (i.e., reads) each overt stimulus of the form "n + m" and places n in X and m in Y. Rule (3c') decodes the contents of X at the end of the computation, and corresponds to the "print" instruction in computer programming. Rules (3a), (3b), and (3c), of course, correspond to (2a), (2b), and (2c), respectively.

The importance of reducing the amount of information which must be processed internally can easily be seen by comparing two versions of the usual addition algorithm for adding multi-digit numerals, like "u

(4) (a) Encode (read) "u

(b) Add digits u

In applying the ordinary addition algorithm, we normally ignore the processes involved in encoding the stimulus numerals (e.g., 45 + 64) and decoding (e.g., writing) the numerals corresponding to the sums. Thus, one normally thinks of the rule as operating on pairs of numbers and generating other numbers, called sums. Nonetheless, in working with five year old children, for example, it might be desirable to elucidate the encoding and decoding processes involved. Thus, we might want to consider, for example, the specific processes by which the numeral "5" can be constructed (e.g., Make a horizontal straight line. Go to the left end point and make a short vertical line downward, etc.)
What is a Rule? 31

(c) If one or more unadded digits remain, do Rule (b); otherwise do Rule (d).
(d) Decode the various digits in I in appropriate order (i.e., write the sum).
(e) If contents of location X = 1, do Rule (f); otherwise do Rule (g).
(f) Decode 1 to left of sum.
(g) STOP.

(5) (a) Encode "u_i" and/or "v_i" of lowest index, which have not yet been decoded (and store 0 in carry location X).*
(b) Add X, u_i and/or v_i.
(c) Decode units digit of sum (X + u_i + v_i) and replace contents of X with tens digit (0 or 1) of sum.
(d) If there are unread digits, go to (do) Rule (a); otherwise do Rule (e).
(e) If contents of location X = 1, do Rule (f); otherwise do Rule (g).
(f) Decode 1 to left of sum.
(g) STOP.

(In interpreting procedures it is understood that rules are performed in the indicated order unless a branching rule indicates otherwise. Strictly speaking, the phrases in Rules (4a) and (5a) set off in parentheses and marked with an asterisk do not involve encoding. See the discussion below on retrieval (search) rules.)

As can readily be seen by applying both procedures (e.g., to 78,747 + 3,416), the requirements on short-term memory of procedure (4) are much greater than those of procedure (5). In procedure (4), the numbers are stored as wholes and all of the intermediary computations are carried out (and held) in memory before anything is written down (i.e., decoded). Procedure (5) corresponds to the more usual procedure of carrying out the algorithm one step at a time. Both procedures would be equivalent in terms of the (D, O, R) characterization, and both might be called "the" addition algorithm, but any person who has tried each of them would surely notice the difference. Almost any reasonably competent fifth grader could correctly apply procedure (5), but there are very few people indeed who could effectively use procedure (4) with large numbers. (For further discussion of information processing, see Chapter 10).

To summarize so far, four kinds of rules have been identified:
(a) **encoding** rules for determining the critical properties $X_1, \ldots, X_n$ of (overt) stimuli.

(b) **operation** rules for transforming properties,

(c) **branching** rules for testing properties, to determine which operating rule to apply next, and

(d) **decoding** rules for making the products of a computation observable.

Together, operation and decision making rules deal with how information is actually processed. Encoding and decoding rules deal with how external stimulation is entered into the information processor, and how the outputs of such processing are made observable.

These four types, however, do not exhaust all of the possibilities which may be involved in processing information. Any information processor, whether human or machine, has a good deal of information stored in memory, which is potentially available, but which may not enter into the application of a given procedure. Thus, for example, a person's memory of certain key dates in world history, or the formula for finding areas of parallelograms, would probably not enter into the computation of sums. On the other hand, this information might play an important role in organizing a composition concerned with the Renaissance period, or in the computation of areas (of parallelograms).

The main point here is that stored information frequently enters into generating responses so that information processors must have some means available for retrieving such information and making it available to the processing mechanism. We have already seen some examples of this type in discussing the (D, O, R) characterization, where we showed how operators might apply to internal properties as well as to properties of directly observable external stimulation in the environment. Our prime example was where the subject had to retain (or retrieve) the previous response in order to know which box to put an $X$ in on the next trial. Here too, however, the (D, O, R) characterization is too molar to distinguish between procedures in which internally stored and external information are entered into a computation at different points.

In order to make the kinds of distinctions in sequencing and scope which we have found desirable with the other kinds of rules, we can introduce a fifth type, called **retrieving** rules (sometimes called searching rules). Retrieving rules do such things as "putting zeros in particular locations involved in a computation" (e.g., the phrases in Rules (4a) and (5a) marked with asterisks). (Retrieving and decoding rules, of course, may enter into a computation at different points.) Searching rules may also serve equally to retrieve (other) operations. Thus, for example, in
What is a Rule? 33

solving problems, a subject frequently has to retrieve previously learned rules before he can proceed further.

There is also a sixth type of rule. This type is needed for storing information that is being processed in particular locations in long-term memory. Rules of this type have the effect of "tying in" new information with information which has already been stored.

Little is said about either storing or retrieval rules, until Chapter 10 where we sketch a theory of memory that is compatible with the theorizing in Chapters 4 through 9. (Chapters 2 and 3 deal, respectively, with definitional questions and needed mathematical background.)

Even these six types, however, are not sufficiently general for present purposes. In order to obtain maximum generality for our procedures, it is necessary to allow procedures to generate new procedures and, then, to turn around and use these procedures later on (in the computation). This possibility is a direct generalization of the process of "calling" a sub-routine (and using it). In the present case, however, (new) procedures can be derived as well as just called. Notice also that the "outputs" involved in carrying out a procedure of this sort are apt to be "mixed," so as to include both rules (procedures) which are later applied and primitives (which are not). The need for such procedures arises naturally in Chapter 7.

In computer programming, however, only the first six types are commonly used. Procedures (programs) are generally fed into the machine directly rather than generated during the course of a computation.

Furthermore, the role played by the various kinds of rules is taken over in the present theory by the theoretical mechanisms introduced. In Chapter 10 for example, storage and retrieval rules are treated as separate entities rather than as components of other rules. The role they play is usurped by theoretical mechanisms which specify precisely the conditions under which control will shift from one type of rule to another. The learning mechanism introduced in Chapter 7 serves a similar purpose in generating new procedures (and then using them).

Finally, we just note that the procedures discussed throughout this treatise are all of the deterministic variety. That is, a specific operation is specified at each stage of a computation. For certain purposes, it is convenient to relax this requirement so that any one of two (or more) operations may be performed at a given stage. Such procedures are called nondeterministic. Nondeterministic procedures provide a convenient language to use wherever it is difficult (if not impossible) to specify precisely the conditions that determine which of two alternative operations is
performed. (For details see Scott, 1967.) They play a major role in artificial intelligence and in a new project of mine on mathematical problem solving.

2.3 Schematic Representation of Procedures and Hierarchies

As suggested above, it is possible to represent any procedure in terms of a flow diagram. Procedure (2) of Section 2.2, for example, can be represented

![Flow diagram](image)

Figure 1

where the diamond shape represents decision rule (2a) and the rectangles, operating rules. The encoding and decoding rules of procedure (3) can be added to the scheme directly, as shown in Figure 2. (A different shape could be introduced to distinguish between operating rules, on the one hand, and decoding and encoding rules, on the other, but there is no real advantage in doing this for present purposes.)

It is sometimes useful to represent the essential aspects of flow diagrams more simply in terms of directed graphs, where decision rules are represented by points, and operating (and encoding and decoding) rules by arrows. The flow diagrams in Figures 1 and 2 can be represented in terms of the directed graph

![Directed graph](image)

Notice here that single arrows are used to connect points even where more than one operating rule intervenes in the corresponding flow diagram.

In more complicated flow diagrams, it is frequently necessary to "collapse" over sequences of (frequently binary) branching rules in order
What is a Rule?

Figure 2

that arrows have an unambiguous interpretation. Consider, for example, the flow diagram of procedure (5) for adding numbers, as shown in Figure 3. The standard way to represent such a flow diagram (as a directed graph) is

The problem with this kind of representation is that the arrows marked m do not represent operating rules, but rather flow of control—e.g., that decision rule (5d) is to be followed directly by decision rule (5e). Such ambiguity can be overcome by allowing multiple branching at each decision point and labeling start and stop. This gives the directed graph

where the center point has three arrows emanating from it. From now on, we use the latter form of representation exclusively.
Hierarchies of Procedures: A given procedure can be represented at essentially any level of detail that might be desired. In particular, the steps of a procedure may be broken down into what might be called subprocedures (paralleling subroutines). The steps of subprocedures, in turn, may be broken down further in sub-subprocedures. In fact, this process may be continued until the operating rules are nothing more than simple associations and the decision rules are simple discriminations.

It is sufficient here to note that Rule (5b) above (i.e., Add X, u_1 and/or v_1) presumes that the information processor "knows" the equivalent of the basic addition table. And this is equivalent to 100 discrete associations of the form a, b \rightarrow (a + b) (e.g., 5, 4 \rightarrow 9). In effect, Rule (5b) can be replaced by these 100 associations, together with a corresponding decision rule, which amounts to a multiple classification scheme (If 5, 4, do (5, 4 \rightarrow 9); if 0, 1, do (0, 1 \rightarrow 1); etc.). The other operating, encoding, and decoding rules may be similarly broken down. (A more
What is a Rule?

formal argument is given in *Structural Learning II* and Chapter 5.5

Representation of procedure (5) in terms of associations would be unwieldy, so we consider binary addition for illustrative purposes. Encoding and decoding rules are not considered except in passing.

Flow diagram and directed graph representations at the molar (D, 0, R) level are simply

![Flow diagram](image)

and (1) A (2), respectively.

Ignoring encoding and decoding, an intermediate level of detail may be represented as shown in Figures 4 and 5, where the two branching rules correspond to "no carry" and "carry," respectively.

Breaking the two branching rules (a) and (c) and operating rules (e) and (f) down further to multiple classifications and associations, respectively, we get (ignoring START and STOP) Figure 6. The numerals at the tail of the arrows in Figure 6 indicate the inputs of each association and those at the heads, the corresponding outputs. Note also that the outputs depend not only on the inputs but also on the decision rule from whence the operating rules (associations) emanate. For example, 11 elicits 0 when emanating from the "no carry" decision rule, but 1 when emanating from "carry." Representation in terms of associations appears to be particularly useful in dealing with information processing capacity (cf. Chapter 10.)

Any (finite) number of levels of detail can be represented simultaneously by simply replacing each operating rule at each level by a directed graph of that rule at the next level of detail. This process may be

---

5 An equivalent way of conceptualizing rules in terms of procedures is described in Scandura (1970c). The basic argument parallels that of Suppes (1969a) and Arbib (1969) and shows how rules can be characterized in terms of networks of associations on associations (in which the latter serve as stimuli). In this case the previous response helps to determine the next response (operating rule). It may also be worth noting that certain of the "associations" in Figure 3 bear a strong similarity to what have traditionally been called concepts. For example, notice first that in encoding according to Rule (5a) the digits in all columns except one are irrelevant (attributes). Hence, this rule would have to be broken down, not into 100 different stimuli, but rather 100 different equivalence classes of stimuli. For example, reading the 4 and 7 in +67 would involve precisely the same association as would reading the 4 and 7 in +273.
Figure 4

(a) If "next" numerals are 1 and 1, do (b); otherwise do (f), unless no numerals, then STOP

(b) write "0"

(c) If "next" numerals are 0 and 0, do (d); otherwise, do (e)

(d) write "1"

(e) write smallest digit

(f) write largest digit

STOP

Figure 5

Start (3)

B

Stop

C

Start (4)
What is a Rule?

Figure 6

or

Figure 6

Figure 7
continued down to the level of associations. For example, the three procedures above for doing binary addition can be represented by the hierarchy shown in Figure 7. In order to facilitate interpretation of this hierarchy, numbers are used to designate corresponding decision rules (sans START and STOP). Capital letters refer to operational rules (boxes in the hierarchy) at one level which are replaced by directed graphs at the next level of detail.

Two further observations are worth comment in passing. First, the associative networks used to represent rules correspond precisely to diagrams used to represent finite automata. Figure 7, for example, is essentially that given by Minsky (1969, 23). The only difference is in terminology; what we have called decision rules correspond to states of an automaton.

Encoding and decoding rules, however, are not usually considered in automata theory. Everything is assumed to enter and exit from automata automatically via tapes on which there are sequences of zeros and ones. These tapes are assumed to move one entry at a time and the machine automatically "reads" or enters digits onto the tape in a strictly mechanical fashion so any specific mention of these mechanisms would be redundant. Notice also that automata are only useful in representing procedures at what might be called the ultimate level of detail. In view of Suppes' (1969a) recent result connecting associations and automata (also see Arbib, 1969), it should come as no surprise to the reader familiar with this result that representing rules as automata is essentially the same as representing rules in terms of associations (and associations on associations) (Scandura, 1970c).

The second observation is that TOTE hierarchies, which were first introduced by Miller, Galanter, and Pribram (1960), are similar to our graphical way of representing rule hierarchies. In TOTE hierarchies, however, no distinction is made schematically between arrows which represent operation rules (operators) and arrows which simply indicate that one decision rule (test) is to be followed by another. Further, some decision rules are identified when they are not the same, while others are distinguished schematically even though they are identical.

This leads to problems in representing rule hierarchies such as that above, where certain of the points (e.g., (3) and (4)) must be distinguished in order to retain the essential structure of the procedure, whereas others (e.g., those labelled (5) and (6)) should not be. Thus a TOTE representation (cf. Miller & Chomsky, 1963, p. 487) of the above hierarchy might
look like

Figure 8

Notice that the horizontal arrows from (3) to (4) and (4) to (3) refer to operational rules whereas the horizontal arrows which connect the points (5) and the points (6) simply represent a transfer of control from one decision point to another (all of which are equivalent operationally). All of the "looped" arrows, however, do represent operational rules as they should.

In effect, TOTE hierarchies force things into a certain form and thereby may destroy essential aspects of the structure of a rule hierarchy. In general, they adequately reflect the structure only of simple hierarchies which involve only loops within loops.

3. BEHAVIOR POTENTIAL

Two major assumptions of the previous discussion are that no matter how many different responses are involved, or whether the eliciting stimuli are overt or internal: (1) all behavior is generated by rules, and (2) rules can be devised to account for all kinds of human behavior. It was also assumed, only this time implicitly, that observer-experimenters always attend just to those behaviors which given subjects happen to be generating. No consideration was given to the problems which might arise when this is not the case—when the behavior the experimenter has in mind and that generated by a given subject are not compatible.
Irregularities will generally be observed unless the subject is using a rule which generates the experimenter-fixed criterion (i.e., class of behaviors) precisely. For example, an experimenter might be concerned with whether or not a subject can add (any pair of numbers), whereas the rules the subject knows may generate only certain of the sums, say, those which do not involve carrying. The subject is using a rule, all right, but it is a rule which is not entirely compatible with the criterion behavior specified by the experimenter. Thus, while the subject's behavior from his own standpoint may be entirely regular, it will not necessarily be so from the standpoint of the observer. In effect, relative to any given class of rule-governed behaviors (class of S-R pairs), a subject may be "successful" on some (S-R) instances of the class, but not others.

Predicting behavior, then, is a relative matter, relative to a given class of rule-governed behaviors. As we shall see in Section 3 of Chapter 7, such predictions may be based on rules introduced to account for just those S-R instances in given rule-governed classes which a subject is capable of. It must be cautioned, however, that explanation relative to the theory which follows (in Chapters 7-10) involves more than just accounting for behavior in terms of rules. The basic mechanisms which govern performance and learning refer directly to the knowledge (i.e., rules) had by individual subjects.

The purpose of Section 3 is twofold. First, we show that the behavior potential of a subject relative to a given rule-governed class (corresponding to what is being observed) can be characterized in terms of categories and functors. While not crucial to the subsequent development, this characterization may be useful in more formal treatments of the theory, and so may be of interest to the mathematically inclined reader. (There is a certain degree of artificiality about the characterization, however, and I am not as enamored of the idea as I once was.) The second major purpose is to discuss some of the problems involved in constructing rules to account for behavior where none of the rules immediately available to the observer is adequate. In this case, the observer has somehow to establish a fit between how he feels a given class of tasks should be tackled and how a subject actually tackles them.

In Chapter 7, we show how the behavior potential of a subject relative to a given rule-governed class may be determined via a finite testing procedure. We also show how to construct systematically a rule which generates precisely that behavior. Throughout this discussion, as well as in Chapters 7 thru 9, it is assumed that the subject is unlimited in his ability to process information. That is, he is assumed able to use whatever rule he
What is a Rule? 43

is using (or is attributed to him) without error, and within the (liberal)
time limits imposed by the observer. For example, insofar as our discus-
sion goes, a subject is equally as capable of adding large numbers as
small ones, so long as he "knows" a rule which applies (to both).

In Chapter 10, we "enrich" our theory to deal with the case where the
subject has a limited information processing capacity. Although we shall
not explore the implications, it may be worth knowing that categories and
functors may be used equally as well to represent behavior potential re-
lative to a given rule-governed class, where the subject has a limited in-
formation processing capacity, as well as where this capacity is unlimited.

3.1 Formal Representation of Behavior Potential in Terms of Categories
and Functors

In order to represent behavior potential relative to a given rule-
governed class, we could simply introduce the notion of a partial function,
as is done in recursion theory and automata theory, and let it go at that.
(A partial function is simply a function which is defined on only part of
its domain.) This would be okay as far as it goes, but we have preferred
instead to borrow the even more fundamental mathematical notions of cate-
gories and functors. These latter notions allow a more detailed character-
ization of behavior potential and, in particular, they lend themselves
quite naturally to operational definition in terms of behavior. (The
emergence of categories and functors has been a relatively recent develop-
ment in mathematics. A growing number of mathematicians (e.g., Freyd,
1964), in fact, feel that these ideas are even more fundamental than sets
and, as such, their use may result in a complete reorganization of the
entire mathematical edifice. To some extent, this has already happened in
homological algebra and algebraic topology where "category theory" has
proved to be a very useful tool in analyzing interrelationships between
different mathematical systems.)

Categories and Functors: Loosely speaking, a functor is a function
between two categories (e.g., a stimulus category and a response category)
which preserves the category structures. Paralleling Freyd's (1964) for-
mal treatment, a category $\mathcal{C}$ may, as an approximation, be viewed as a class
of objects with a structure. As in the previous sections, the objects in
the classes may be viewed as equivalence classes of overt stimuli and
responses (effective stimuli and responses) without loss of generality.
By structure on the classes is meant that transformations, or maps, between

Footnote on following page.
the objects are subject to the following constraints: (1) If a map $x$ exists from $A$ to $B$ (where $A$ and $B$ refer to equivalence classes of overt stimuli) and $y$ exists from $B$ to $C$, then a map, $z = xy$ (read, applying $z$ is equivalent to applying $x$ and then $y$) exists from $A$ to $C$. All this can be represented schematically as: $A \xrightarrow{z} B \times C$. (2) For any three maps, $A \xrightarrow{x} B$, $B \xrightarrow{y} C$, and $C \xrightarrow{z} D$, applying map $A \xrightarrow{xy} C$, followed by $C \xrightarrow{z} D$ is equivalent to applying $A \xrightarrow{z} B$ followed by map $B \xrightarrow{yz} D$—symbolically, $(xy)z = x(yz)$ which corresponds to the familiar associative property of algebra. (Note: To simplify the discussion below, we do not require identity maps (from objects to themselves) to exist for all objects.)

For present purposes, a functor, $F$, may be viewed as a function between a stimulus category, $\mathcal{A}$, and a response category, $\mathcal{R}$, which transforms (i.e., maps) each object $A$, in $\mathcal{A}$, into an object, $F(A)$, in $\mathcal{R}$, and which transforms each map in $\mathcal{A}$ into a map in $\mathcal{R}$ so that:

(a) if maps $A \xrightarrow{x} B$ and $B \xrightarrow{y} C$ exist in $\mathcal{A}$ (so that $xy$ from $A$ to $C$ also exists), then the corresponding maps $F(x)$ and $F(y)$ in $\mathcal{R}$ go from the objects $F(A)$ and $F(B)$ to $F(B)$ and $F(C)$, respectively, so that $F(xy) = F(x)F(y)$ (i.e., the map $xy$ from $A$ to $C$ is transformed into a map $F(xy)$ in $\mathcal{R}$ which is identical to the transform, $F(x)$, of $x$ which goes from $A$ to $B$, followed by the transform, $F(y)$, of $y$ which goes from $B$ to $C$); 7, 8 (b) If $A \xrightarrow{e} B$ is an identity map in $\mathcal{A}$, then $F(e)$ is an identity map in $\mathcal{R}$.

6 Maclane and Birkhoff (1967) have recently made the idea of a "concrete category" available to non-mathematicians. This notion corresponds to Freyd's (1964) "first approximation" to the definition of a category which Freyd used simply to motivate his more abstract treatment. Although they do treat category and functor theory as such, Maclane and Birkhoff (1967) were generally more concerned with the application of these ideas to abstract algebra (which, oddly enough, is less "abstract" than are category theory and functor theory).

7 In contrast to Freyd (1964), I have written composite maps in the diagrammatic order (i.e., the order in which they are applied). Thus, $xy$ ($F(x)F(y)$) is the map which is equivalent to first applying $x$ ($F(x)$) and then $y$ ($F(y)$). On the other hand, when discussing functor values I have used the traditional (i.e., reverse) order—letting $F(x)$ refer to that map in $\mathcal{R}$ obtained by applying $F$ to $x$ (in $\mathcal{A}$) and letting $F(A)$ refer to that class in $\mathcal{R}$ obtained by applying $F$ to $A$ (in $\mathcal{A}$). I hope this will not prove confusing.

8 It is no accident that the most important features of these definitions involve maps rather than the objects the maps are defined on. In fact, categories and functors are defined most generally without any reference to the objects themselves (Freyd, 1964). In this case, each object is replaced by an identity map (which takes the object onto itself). (continued on next page)
Behavior Potential and Functors: There is obviously a close relationship between classes of rule-governed behavior and functors between corresponding stimulus and response categories. Both associate with each (effective) stimulus a unique (effective) response. (The effective stimuli and responses, of course, are the objects in the categories.) In addition, functors transform maps in stimulus categories into maps in response categories. This dual role makes it possible for functors to represent simultaneously not only given classes of rule-governed behavior but the behavior potential of individual subjects relative to the given classes as well. Specifically, for each class of rule-governed behaviors, there are any number of functors which transform the stimulus objects in a corresponding stimulus category into response objects in a corresponding response category precisely as specified by the given class of behaviors. What the introduction of maps between the objects in the categories does is to make it possible to represent the behavior potential of individual subjects relative to the given rule-governed class.

To see how this can be accomplished, it is sufficient to note that functors may be represented by pairs of functions, one which maps stimulus objects in one category into response objects in another, and a second which takes maps in the stimulus category into maps in the response category. The first function corresponds directly to the given rule-governed class (i.e., it takes stimuli into corresponding responses). The second function corresponds to that subset of the rule-governed class which the subject in question can actually generate.

The objects in the categories are obviously the stimuli and responses, respectively, which are associated with the given rule-governed class. It is not immediately clear, however, just what the maps correspond to and a certain degree of artificiality enters at this point. Nonetheless, maps may be said to exist between a pair of (distinct) stimuli if the subject has a learned rule available which applies to each stimulus in the pair. On the response side, maps are said to exist between those pairs of

(continued) A category, then, is simply a class of maps (between "imaginary" objects) together with a specified subclass of pairs of maps such that the composite map which is equivalent to the first map of the pair followed by the second is in the class of maps. The pairs of maps together with the "associative" restriction (see above) effectively determine the structure of the category. The identity maps make it possible to recover (i.e., specify) the original objects if this is desired.

Dropping the objects and replacing them with identity maps, results in only minor changes in the above definition of a functor. Again following Freyd (1964), a functor from category $\mathcal{A}_1$ to $\mathcal{A}_2$ is a function $F: \mathcal{A}_1 \to \mathcal{A}_2$ such that: (1) If $e$ is an identity map in $\mathcal{A}_1$, then $F(e)$ is an identity map in $\mathcal{A}_2$ and (2) If $xy$ is defined in $\mathcal{A}_1$ then $F(x)F(y)$ is defined in $\mathcal{A}_2$ and is equal to $F(xy)$. 

responses given by the subject which are "correct" in the sense that they are the ones specified for the various stimuli in the given rule-governed class.

Maps defined in this way automatically satisfy the requirements of category membership. In particular:

(1) If the subject has a learned rule which applies to stimuli $S_i$ and $S_j$, and another (possibly the same) to $S_j$ and $S_k$, then he necessarily has a rule which applies to $S_i$ and $S_k$. Similarly, if responses $R_i$ and $R_j$ are correct, and $R_j$ and $R_k$ are correct, then $R_i$ and $R_k$ are correct. These observations correspond to the category requirement that the existence of two maps, $S_i \rightarrow S_j (R_i \xrightarrow{F(x)} R_j)$ and $S_j \rightarrow S_k (R_j \xrightarrow{F(y)} R_k)$, implies the existence of the composite map, $xy \xrightarrow{F(xy)}$.

(2) For any four stimuli, $S_i$, $S_j$, $S_l$, $S_k$, a learned rule is available for $S_i$ and $S_k$, whenever there are learned rules available for $S_i$ and $S_j$ and for $S_j$ and $S_k$ or for $S_i$ and $S_l$ and $S_k$. On the response side, "correct" pairs of responses act in precisely the same manner. These observations correspond to the category requirement (associativity) that for any three maps $S_i \rightarrow S_j (R_i \xrightarrow{F(x)} R_j)$, $S_j \rightarrow S_k (R_j \xrightarrow{F(y)} R_k)$ and $S_k \rightarrow S_l (R_k \xrightarrow{F(z)} R_l)$, $xy \xrightarrow{F(xy)}$ followed by $z \xrightarrow{F(z)}$ is equivalent to $x \xrightarrow{F(x)}$, followed by $yx \xrightarrow{F(yx)}$. (Notice that our definitions say nothing about identity maps which are ordinarily assumed to be a category requirement. For this reason we are unable to distinguish between individual stimuli (and responses) for which associations do and do not exist.)

In general, this definition requires that maps exist between all of the stimuli and all of the responses, respectively, which are associated with just those instances of the rule-governed class on which a given subject is successful. Any particular pattern of behavior on the instances of a given rule-governed class, then, uniquely specifies a functor which represents it. (More precisely the pattern of behavior specifies the functor up to an equivalence class in which isolated associations are ignored). Conversely, of course, any given functor specifies a unique pattern of behavior. In particular, successful performance on all instances of the given rule-governed class corresponds to a functor between two categories having a "complete" structure in the sense that maps exist between each pair of objects in the respective categories. Complete failure corresponds to a functor between structureless categories (with no maps).

Of course, it will generally be impossible (or impractical) to observe (test) performance on all instances of a given class. Hence, any attempt to determine the behavior potential of an actual subject will necessarily have to depend on what assumptions can be made about human performance.
What is a Rule?

(The category requirements do not provide any means for going beyond behavior which is observed directly.) Coming up with a functor which accurately represents an individual's behavior potential, then, will necessarily require positing the existence of untested maps on the basis of available information.

Detailed discussion of this problem is postponed until Chapter 7 but the general idea behind the proposed solution may be seen in the following example. Consider the task of giving the sum of (long) arithmetic number series—that is, number series in which the consecutive numerical terms differ by a constant. We found this to be a useful task in some of our earlier experiments (Scandura, Woodward & Lee, 1967) because the stimuli vary over three independent and well-defined dimensions: (1) value of the initial term (A), (2) value of the common difference (D), and (3) the number (N) of terms. Each arithmetic number series can be uniquely characterized by a particular triple of values associated with these three variables. Thus, for example, two stimuli might have the same initial term and the same common difference, but a different number of terms (e.g., 1 + 3 + 5 + ... + 49 and 1 + 3 + 5 + ... + 63). There is a well-known formula, \[(A + L)/ 2N\], where the last term \(L = A + (N - 1)D\), by which the sum of any arithmetic number series can easily be computed.

On this class of tasks, suppose we find that a particular subject is consistently able to sum number series of the form 1 + 3 + 5 + ..., but that he is unable to sum arithmetic number series which have a different first term and/or common difference (except by sequential addition which we somehow disallow—e.g., by imposing a time limit on the answer). The series 2 + 6 + 10 + ... + 42 is one such series.

After observing this pattern repeatedly, say, five or six times in a row, we would probably be satisfied that we knew something about what kinds of behavior the subject was and was not capable of. We would simply stop testing—confident that we could predict his behavior on any other arithmetic series problem.

Correspondingly, we undoubtedly would be willing to postulate the existence of maps between arbitrary stimuli which differ along the "number of terms" dimension but, at the same time, would not want to assume the existence of maps between stimuli which differ along either the first term or common difference dimensions. Of course, because of the structure preserving properties of the functor characterization, corresponding maps would be postulated in the response category as well. In Chapter 7 we shall see that the very nature of a dimension in the psychological sense is tied very closely to the nature of the procedures introduced to account
for the behavior. Positing maps between stimuli (responses) essentially amounts to making assumptions about procedures which might account for the behavior. Specifically, the behavior a subject is capable of can be accounted for in terms of particular paths through a procedure which accounts for the given class of rule-governed behaviors.

3.2 Accounting for Behavior Potential

The rule(s) an experimenter has in mind for generating a class of rule-governed behaviors and the rule(s) a subject might use on the corresponding tasks are not necessarily the same. In particular, the rule(s) the subject uses may be adequate to generate certain behaviors in the class but not all of them. Accounting for the subject's behavior potential, then, amounts to coming up with a procedure which accounts for just those behaviors which the subject is actually capable of. The accounting procedure is simply a construct introduced by an observer to explain a subject's behavior and may have little directly to do with the rule(s) the observer had in mind originally for generating the given rule-governed class (or for that matter with the rule(s) the subject actually uses).

We reemphasize this point because of the basic differences of opinion among present-day theoreticians as to whether rules actually "exist" in people's minds or whether these constructs are to be thought of simply as ways of explaining behavior. According to the present view, this is not an either-or question. Subjects do learn and do use rules, and as we shall see in Chapters 7-9, our theoretical hypotheses apply directly to these rules. It is also true, however, that observers may introduce rules to account for classes of rule-governed behavior, where these classes may either be given on a priori grounds or represent an observed repertoire of actual behavior.

Coming up with procedures, which account for the actual behavior potential of particular individuals, typically requires more ingenuity on the part of an investigator than accounting for given classes of rule-governed behaviors since the latter usually correspond to well-known procedures of various kinds. When a subject's behavior is "imperfect" relative to the given rule-governed class, the investigator must somehow put himself in the position of the naive subject, and this is sometimes quite difficult.9

9Readers having some acquaintance with educational measurement may recognize the similarity between this problem and that of diagnosing a child's source of difficulty on a class of tasks—that is, the problem of diagnostic testing. Diagnostic testing is particularly common in the area of reading and, in my opinion, should be used more frequently (continued)
To gain some insight into the problem, consider, for example, the behavior which might be accounted for by an "imperfect" version of the procedure, denoted \((\frac{A+L}{2})N\), for summing arithmetic number series. Suppose that the subject in question is only able to determine \(N\) (the number of terms) for series in which the first term \((A)\) is one and the common difference between adjacent terms is two—that is, number series of the form \(1 + 3 + 5 + \ldots + (2N-1)\). Suppose he is also able to "encode" the first \((A = 1)\) and last \((L = (2N-1))\) terms of such series. Then, assuming that the other steps in the procedure are intact, the kinds of behavior the subject can generate can be completely specified; these behaviors correspond precisely to that subclass (of series) whose sums can be generated by the procedure formed from \((\frac{A+L}{2})N\) by restricting it to \((\frac{1+L}{2})N = N^2\) (i.e., substituting \(A = 1\) and \(L = 1+(N-1)D\), where \(D = 2\)). The number series to which this restricted procedure applies are all of the above form. In effect, the behavior potential relative to the class of arithmetic series consists precisely of that subclass in which the stimuli may vary only over the dimension \(N\); \(A (=1)\) and \(D\) (the common difference between adjacent terms = 2) are constant.

Other kinds of "flaws," of course, might be found in other steps of a procedure. In adding numbers, for example, a subject who lacks the "carry" rule would be quite unable to add numbers in which carrying is involved. An example of a decoding inadequacy is provided by a child who is unable to write certain numerals.

Unfortunately, there is no mechanical procedure by which one can devise a procedure which accounts for the behavior potential of an arbitrary subject. Nonetheless, it is usually possible to devise such a procedure in the following manner. First, identify all (as many as possible) of the rule(s) which generate the given rule-governed class and which correspond to how the subject(s) in question might possibly generate the behaviors. Second, break the(se) rule(s) down into procedures at as gross a level as might reasonably account, if suitably restricted, for the behavior potential. Third, see if the behavior potential can be accounted for precisely by removing any of the individual rules in (any one of)

In mathematics education. There are far too many people who are "turned off" to mathematics for a lifetime because of inattention to this problem during the early school years. The problem is equally crucial at various other stages through which students may pass in their mathematical education. One such stage corresponds to the shift from a computational emphasis, like that of the traditional calculus course, to more axiomatic approaches which students frequently first encounter in studying "abstract" algebra.

If even one such rule exists, then a countably infinite number of other rules also exist.
the(se) procedure(s) (or some combination thereof). If this doesn't work, break down the procedure(s) into still less molar procedures and try again. This process can be repeated as many times as necessary, or until the original rule(s) is (are) represented entirely in terms of associations (see above).

It should be emphasized, however, that no matter how far a given rule is broken down, it is always possible to conceive of certain patterns of behavior for which it simply cannot account. Specifically, the algorithmic (rule) account of behavior proposed (also see Chapter 7) disallows the possibility of being able to use a given path of a procedure to generate one behavior (S-R pair) in the class but not being able to use it with another behavior. A subject who can carry on one addition problem, for example, is assumed able to carry on any other. Component rules, in effect, are assumed to act on an all-or-none basis.

Coming up with an adequate account will generally require some degree of ingenuity both in selecting rules in the first place and in knowing which component rules to refine further, and which to leave alone.
In this chapter, we introduce a number of mathematical preliminaries which are utilized in Chapter 6. There, we consider the central question of what it means to know mathematics. The reader with a good grasp of mathematical foundations may wish to skip this chapter entirely. In the present approach, however, algebraic systems are defined in terms of concrete embodiments, and it is probably advisable to at least skim this chapter before going on.

1. CONCRETE EMBODIMENTS

Although itself abstract, mathematics derives ultimately from the real world. Indeed, mathematics is frequently modeled directly on concrete reality; and, where it is not, mathematics is modeled on previously developed mathematics (e.g., Euclidean n-space was motivated largely by geometry of the plane and 3-dimensional space).

In the present section, we consider the nature of concrete embodiments. Concrete embodiments involve a basic set of objects (e.g., triangles), actions (e.g., rotations), or relationships (e.g., kinship) and one or more operations and/or relations defined on this basic set, together with possibly one or more distinguished elements of the set.

More precisely, what we shall call mathematical embodiments can be characterized as n-tuples, \( \langle A; 0_1, 0_2, ..., 0_n; R_1, ..., R_m; e_1, ..., e_p \rangle \), where \( A \) is some basic set of elements, the \( 0_i \) (\( i = 1, ..., n \)) and the \( R_j \) (\( j = 1, ..., m \)), respectively, are operations and relations defined on the basic set, and the \( e_k \) (\( k = 1, ..., p \)) are "distinguished" elements of the basic set. The only additional requirement is that the basic elements, operations, and relations be concrete.

Even so, embodiments are already abstractions of reality in the sense that certain idiosyncratic features of the elements, and the operations
and relations defined on them, are ignored. Specifying a rotation of 90°, for example, says nothing about either the exact manner in which the rotation is performed physically or even the object being rotated. We consider a number of examples.

Example 1: Consider the set of three symmetry rotations of an equilateral triangle, that is, the 0°, 120°, and 240° rotations which map the set of positions of a given equilateral triangle onto itself. (Each of the rotations, of course, is a function which can be characterized as a set of ordered pairs. The first element of each pair is a position before the rotation and the second element is the corresponding position after the rotation.)

In addition to a basic set of elements, which in this case consists of the three rotations, a mathematical embodiment must involve at least one operation or relation defined on this set. In the present situation, it is perhaps most natural to introduce the binary operation of composition. This operation associates an arbitrary pair of given rotations with a new one called the composite rotation. The operation may be defined as follows: Given rotations A and B, then the composite rotation A o B (say, "A followed by B") is equivalent to that single rotation which results in the same action as first performing rotation A and, then, performing rotation B (e.g., 120° o 240° = 0°). Such operations are denoted by (A, B) → A o B. Operations of this sort may be defined in a wide variety of different embodiments, and all typically go under the label "followed by." We shall sometimes adhere to this terminology in our subsequent discussions, but it should be emphasized that an operation called "followed by" in one embodiment is not the same as an operation called "followed by" in a different embodiment. The domains differ.

Finally, we notice that a rotation of 0° plays a special role in the embodiment and, thus, counts as a "distinguished" element. In particular, whenever a given rotation A is combined with one of 0°, the result is equivalent to that same rotation (i.e., A).

As indicated above, all this can be summed up by characterizing the embodiment in terms of the three-tuple (A, o, 0°) where

\[ A = \{0°, 120°, 240°\} \] and
\[ o = \{(0°, 0°) \rightarrow 0°; \ (0°, 120°) \rightarrow 120°; \ (120°, 0°) \rightarrow 120°; \ (0°, 240°) \rightarrow 240°; \ (240°, 0°) \rightarrow 240°; \ (120°, 120°) \rightarrow 240°; \ (240°, 240°) \rightarrow 120°; \ (120°, 240°) \rightarrow 0°; \ (240°, 120°) \rightarrow 0°\} \]

*A binary operation may be defined as a mapping between a subset of the cartesian product set A x A of a given set A into the original set A.*
Example 2: A second example may be obtained by complicating the embodiment of Example 1 by allowing additional actions to be performed on the triangle—say, "flips" about the vertical axis. (Flips about the other axes may be obtained by composition with the rotations.)

In this embodiment, it turns out that the three symmetry rotations together with the basic flip can be combined to form six distinct actions which in this case are all symmetries of an equilateral triangle. The six basic symmetries are as follows: $0^\circ, 120^\circ, 240^\circ, F$ (i.e., a flip about the vertical axis), $120^\circ \circ F$ (a rotation of $120^\circ$ followed by $F$), and $240^\circ \circ F$.

The binary operation defined on these basic elements is a simple extension of that in Example 1 and its definition is left as a simple exercise. (Suggestion: Construct a square array as in an addition table. The sums (i.e., composite symmetries) may be determined as above by performing one of the basic symmetries after the other and seeing which of the original six symmetries is equivalent to it. For example, $(120^\circ \circ F)$ followed by $(240^\circ \circ F)$ is equivalent to a rotation of $240^\circ$.) Again, of course, $0^\circ$ is the distinguished element.

In effect, the embodiment can be characterized as the three-tuple $(A', o', 0^\circ)$ where $A'$ consists of the six basic symmetries and $o'$ corresponds to the "addition" table described above.

Example 3: In the first two examples (above) the basic elements are rotations and not the objects (i.e., triangles) the rotations act on.

The present example shows that the basic elements may act at an even higher level. In particular, the basic elements may be operations which map an embodiment, like that of Example 1, onto itself. One such operation maps $0^\circ$ into $0^\circ$, $120^\circ$ into $240^\circ$, and $240^\circ$ into $120^\circ$. Call it $T_1$. Another ($T_2$) takes $0^\circ$ into $120^\circ$, $120^\circ$ into $240^\circ$, and $240^\circ$ into $0^\circ$. A third possibility ($T_3$) is to simply map each rotation into itself (e.g., $120^\circ$ into $120^\circ$).

Certain of these operations have the interesting property that whatever one does in the embodiment of Example 1 is reflected under the operation in the image. More precisely, the operation takes rotations in the embodiment into other (possibly the same) rotations in the same embodiments in such a way that if the rotations, say, $0_1$, $0_2$, and $0_3$, are related, as indicated by $0_1 \circ 0_2 = 0_3$, before the operation is performed, then doing the rotation associated, under the operation, with $0_1$ followed by the rotation associated with $0_2$, is equivalent to the rotation associated with $0_3$. Letting $T$ be such an operation, this requirement may be represented
The second operation above (T\(^2\)) does not have this property but the first and third operations do. Such operations are called automorphisms. In the first automorphism, for example, notice that 120°, 240°, and 0° are mapped into 240°, 120°, and 0°, respectively. Since 120° o 240° = 0° is the same as \(T(120°) \circ T(240°) = 240° \circ 120° = 0°\), we see that the desired property holds in this particular case. (The reader who is unfamiliar with this idea may want to check the other combinations to make sure that \(T_1\) is indeed an automorphism. The same should be done for \(T_3\).

\(T_2\) is not an automorphism because, for example, 0° = \(T(240°)\) = \(T(120° \circ 120°)\) = \(T(120°) \circ T(120°) = 240° \circ 240° = 120°\); doing 120° twice is the same as doing 240° but 240° is not equivalent to doing the image of 120° (i.e., 240°) twice. The latter equals 120° which is not the image of 240°.)

The point of all this is that the two automorphisms \(T_1\) and \(T_3\) can be thought of as elements in the basic set of a third embodiment. In this embodiment, the binary operation of interest is composition of automorphisms, the automorphisms being thought of as states in this case rather than as actions. The distinguished element, of course, is the "identity" automorphism, \(T_3\).

**Example 4:** The three examples above are all algebraic in nature.

We now consider a simple geometric embodiment which can be represented by the display

![Display Diagram](image_url)

The question is how we are to characterize the embodiment inherent in this display. (While the embodiment must take account of all central aspects of the display, it is not identical with it.)

The first thought that comes to mind might be to characterize the embodiment in terms of the set of five points, the set of five lines, and
the relation between them (i.e., of points being "on" lines). In this case, we would get the three-tuple \( \langle P, L, \text{"on"} \rangle \) where \( P = \{ P_1, P_2, \ldots, P_5 \} \), \( L = \{ L_1, L_2, \ldots, L_5 \} \) and "on" is the binary relation \( \{ (P_1, L_5), (P_1, L_1), (P_2, L_1), (P_2, L_2), \ldots, (P_5, L_5) \} \).

This characterization certainly accounts for all that seems to be involved in the display, but it deviates from our definition above in that it includes two sets, each of which appears to be equally basic. Fortunately, this matter can be satisfactorily resolved by simply introducing a new set which includes all of the points and lines and defining this to be the basic set. When looked at in this way, the set of points and the set of lines simply become unary relations (i.e., subsets) defined on this set. In effect, then, the finite geometry may be characterized as \( \langle P \cup L, P, L, \text{"on"} \rangle \) where the elements in the n-tuple are as defined above. It is important to stress in this case that the terms "point," "line," and "on" refer to the common meaning of these terms; they are not undefined.

**Example 5:** As a final example, let us consider an embodiment of the ordinary arithmetic of natural numbers. In this case, the basic set is infinite; it consists of the numerals 0, s(0), ss(0), sss(0), ... \(^2\)

While it would seem reasonable to include at least the operations of addition and multiplication in the characterization of this embodiment, even this is not necessary. Both operations can be defined in terms of the more basic operation of "successor" which maps the basic set of (natural) numerals into itself so that each numeral goes into its successor (e.g., sss(0) goes into ssss(0)). For example, letting \( s \) correspond to the successor operation, it is a simple matter to define addition in terms of \( s \). Thus, \( A + B \) is defined to be the \( B \)-fold successor of \( A \), denoted

\[
\underbrace{s \ s \ ldots s}_{\text{B times}} (A) \quad (s^B(A) \text{ for short})
\]

In view of their special role in addition and multiplication, of course, the numerals 0 and 1 would be distinguished. To summarize, the arithmetical embodiment may be characterized as \( \langle N, s, 0, 1 \rangle \), where \( N = \{ 0, s(0), ss(0), \ldots \} \) and \( s \) is defined as above.

**Equivalence of Relations and Operations:** Although there is distinct heuristic value in distinguishing between operations and relations, it is possible to define embodiments without any reference to operations (and conversely). To see this, notice that in each of the above examples, each operation can be represented as a relation. Thus, for example, the binary

\(^2\)Names of numbers (i.e., numerals) are observable, numbers are not.
operation "followed by" of Example 1 can be represented as the ternary relation \{\(0^\circ, 0^\circ, 0^\circ\), \(0^\circ, 120^\circ, 120^\circ\), ...\}. Similarly, the unary (successor) operation of Example 5 can be viewed as a binary relation. Conversely, of course, relations may be represented as operations. For example, the unary relations of Example 4, by which the subsets of the basic set were formed, can be viewed as 0-ary operations. Distinguished elements may be viewed as 0-ary relations.

In general, any n-ary relation \((n \geq 1)\) can be represented as an \((n - 1)\)-ary operation and vice-versa. Thus, any given embodiment may be redefined as an n-tuple consisting of a basic set and one or more relations. In working in the foundations of mathematics and in logic, there are a number of technical, as well as traditional, reasons for preferring this approach. Some of these reasons are mentioned in a later section.

2. MATHEMATICAL SYSTEMS

The basic entities with which (abstract) mathematics is concerned, of course, are undefined. As Bertrand Russell has put it in an often quoted remark: Mathematics is a subject where we don't know what we are talking about or whether what we are saying is true. More succinctly, mathematics may be called "abstract nonsense." *

The objects of abstract mathematical study are called algebraic systems. Algebraic systems are like embodiments in the sense that they can be characterized in essentially the same way but they are unlike embodiments in the sense that the basic elements, operations and relations have no meaning.

In order to define algebraic systems more precisely, we introduce the notion of equivalence of embodiments. Two embodiments may be said to be equivalent if and only if they are isomorphic in the sense that there is a one-to-one onto map (i.e., a bijection) between the elements of the respective basic sets which preserves the respective operations and relations (including distinguished elements). That is, two embodiments

\[ E = \langle A; R_1, \ldots; 0_1, \ldots; e_1, \ldots \rangle \text{ and } E' = \langle A'; R'_1, \ldots; 0'_1, \ldots; e'_1, \ldots \rangle \]

are isomorphic if (a) there exists a bijection, \(b\), between the basic sets \(A\) and \(A'\); (b) the ordered set \[a_1, a_2, \ldots, a_n \in R \iff [b(a_1), b(a_2), \ldots, b(a_n)] = [b(a_1), b(a_2), \ldots, b(a_n)] \in R' \]

where \(a_1, a_2, \ldots, a_n \in A\); (c) \(0_i(a_1, \ldots, a_n) = 0_i(b(a_1), \ldots, b(a_n)) = 0_i(a'_1, \ldots, a'_n)\) is defined where \(a_1, a_2, \ldots, a_n \in A\); (d) \(b(e_i) = e'_i\) for all \(e_i \in A\).

* I got the phrase from Peter Freyd; where he got it, I do not know.
For example, consider the embodiment of Example 1 together with one involving the set of cyclic permutations on the ordered set \((a, b, c)\) .³ This embodiment may be formally characterized as \(\langle\{0\text{-cycle, 1-cycle, 2-cycle}\}, o'\rangle\) where \(o'\) corresponds to the binary operation "followed by" in Example 1. Clearly, the bijection 

\[
\begin{align*}
0^\circ & \leftrightarrow 0\text{-cycle} \\
120^\circ & \leftrightarrow 1\text{-cycle} \\
240^\circ & \leftrightarrow 2\text{-cycle}
\end{align*}
\]

defines an isomorphism between the embodiments. To see this, notice that 

(a) the distinguished element, \(0^\circ\), goes into the distinguished element, 0-cycle, (i.e., \(b(0^\circ) = 0\text{-cycle}\)) and (b) \(b(R_1 o R_2) = b(R_1) o' b(R_2) = C_1 o' C_2\) where \(R_1\) and \(R_2\) are arbitrary rotations of the first embodiment and \(C_1\) and \(C_2\) are corresponding cyclic permutations of the second. For example, 

\[
b(0^\circ) = b(120^\circ o 240^\circ) = b(120^\circ) o' b(240^\circ) = 1\text{-cycle} o' 2\text{-cycle} = 0\text{-cycle}.
\]

Another pair of isomorphic embodiments is obtained by considering Example 4 together with the embodiment consisting of the basic set 

\[
\{\text{Jeanne, Janie, Joey, Julie, Alice, reading committee, cleanup committee, eating committee, sewing committee, dinner committee}\}
\]

The relations defined on this basic set may be called the (a) people's relation (i.e., the subset of people), (b) committee relation (i.e., the subset of committees), and (c) "being on" a committee relation. To make the discussion definite, we define the third relation to be 

\[
\{(\text{Jeanne, reading}), (\text{Janie, reading}), (\text{Janie, cleaning up}), (\text{Joey, cleaning up}), (\text{Joey, eating}), (\text{Julie, eating}), (\text{Julie, sewing}), (\text{Alice, sewing}), (\text{Alice, dinner}), (\text{Jeanne, dinner})\}
\]

The terms "point," "line," and "on," which refer to particular marks and relationships in the first embodiment, correspond respectively, to people, committees, and being on committees in the second system. In another isomorphic embodiment, the points, lines, and relation "on" might correspond to intersections, streets, and intersections of streets, respectively. Clearly, the number of embodiments which are isomorphic to any given

³A cyclic permutation on an ordered set (i.e., an n-tuple) may be described as "the action" involved in moving every element of the ordered set some number of positions (0 to n-1) to the right (or left) so that if an element is "pushed off" the end, it starts back at the left. For example, a 1-cycle to the right on \((a, b, c)\) would result in \((c, a, b)\); a two-cycle, in \((b, c, a)\); and, of course, a zero (or three-cycle) in \((a, b, c)\).

As was the case with rotations, cycles can be defined as sets of ordered pairs, in particular, as sets of pairs of 3-tuples.
embodiment is indeterminate and will be limited in any particular case only by our ingenuity in devising new ones.

With this background, we might be tempted to define a mathematical system as the class of all isomorphic embodiments having a particular "structure." For such a definition to be meaningful, however, it is necessary to specify the underlying structure. And, to do this, one must either use some meta-language to describe it or "give an example." The latter approach amounts essentially to displaying a particular embodiment, usually one in which the basic elements, operations and relations are "meaningless" symbols, called undefined terms. The characterizing embodiment may be distinguished with the name canonical to indicate its special significance.

In selecting a canonical embodiment (of a mathematical system), of course, it is desirable to choose an embodiment with as few "distractors" as possible. In this sense, the above choice of Jeanne, Janie, Joey, Julie, and Alice would probably be a poor one. The names carry more information than that they belong to certain committees, especially for the author, since the first four happen to be his children and the last, his wife. A better selection, for example, would be to define the algebraic system, corresponding to Example 1, to be \( \{a, b, c\}, a, o \), where \( a, b, \) and \( c \) correspond to \( 0^\circ, 120^\circ, \) and \( 240^\circ \), respectively, and \( o \) to the binary operation, "followed by."

Once a canonical embodiment has been specified, we define the class of embodiments isomorphic to it to be the denotation (meaning) of the system—or more simply, the algebraic system. Notice that the symbols used to characterize some of the embodiments of this meaning class may be no more "meaningful" than those used to characterize the algebraic system itself.

In standard mathematical treatments, mathematical systems are defined first and the basic terms are said to be undefined. What we have called embodiments result on assignment of meaning to these undefined terms. Defining mathematical systems as above has the advantage of distinguishing between expressions denoting mathematical systems (canonical embodiments) and mathematical systems themselves.

### 3. INFORMAL AXIOMATIC THEORIES

Roughly speaking, theories may be thought of as sets of properties of embodiments or algebraic systems. That is, theories are concerned with the descriptive aspects (i.e., properties) of such entities rather than with the entities themselves.

Since embodiments consist of objects, operations, and relations which
are directly observable, properties of embodiments relate directly to reality. Thus, for example, the embodiment of Example 1 has the property: Given any pair of rotations, \( r_1 \) and \( r_2 \), there is a unique rotation \( r_3 \), such that \( r_1 \) followed by \( r_2 \) results in the same action as \( r_3 \). (Example 2 also satisfies this property.) Physical theories deal with such properties.

Mathematical systems, on the other hand, are abstract. They refer to classes of embodiments. Properties of algebraic systems, therefore, are properties of classes of real world objects. (Note: The classes are not themselves in the real world.) In this sense, properties of algebraic systems are like "place holders" or property schemata which hold in (i.e., become properties of) each embodiment of the system on assignment of meaning to the undefined terms. Thus, for example, the property schema, corresponding to the property described above, might be stated, "Given any pair of elements \( A \) and \( B \), there exists a unique element \( C \), such that \( A \circ B = C \)" (where \( A, B, \) and \( C \) are undefined elements and \( \circ \) is an undefined binary operation, which bears the indicated relation to the elements). Mathematical theories, then, are concerned with property schemata of algebraic systems. (In the discussion which follows, we shall distinguish between properties and property schemata only where necessary to avoid confusion.)

For future reference, it will be helpful to list some properties of the embodiments and algebraic systems corresponding to our five basic examples. In each case, we list a number of properties and indicate how the corresponding property schemata of the associated algebraic system relate to them.

**Example 1:**

\[ P_1 \rightarrow \text{For all } r_1, r_2, \text{ there exists an } r_3 \text{ such that } r_1 \circ r_2 = r_3 \]

\[ P_2 \rightarrow \text{For all } r_1, r_2, \text{ and } r_3 \]
\[ r_1 \circ (r_2 \circ r_3) = (r_1 \circ r_2) \circ r_3 \]

\[ P_3 \rightarrow \text{There exists a rotation, } 0^\circ, \text{ such that for all } r \]
\[ r \circ 0^\circ = r \]

\[ P_4 \rightarrow \text{For all } r, \text{ there exists an } r' \text{ such that } r \circ r' = 0^\circ \]

\[ P_5 \rightarrow \text{For all } r_1 \text{ and } r_2 \]
\[ r_1 \circ r_2 = r_2 \circ r_1 \]

\[ P_6 \rightarrow 0^\circ \circ 0^\circ = 0^\circ \]

\[ P_7 \rightarrow \text{For all } r_1 \text{ and } r_2 \]
\[ r_1 \circ r_2 = 0^\circ \text{ implies } r_2 \circ r_1 = 0^\circ \]

\[ P_8 \rightarrow \text{For all } r_1, r_2, r_3 \]
\[ r_1 \circ r_3 = r_2 \circ r_3 \text{ implies } r_1 = r_2 \]
P_9 -- For all r
0° o r = r

P_{10} -- For all r_1, r_2, r_3
r_3 o r_1 = r_3 o r_2 implies r_1 = r_2

P_{11} -- For all r_1, r_2, ..., r_5
(((r_1 o r_2) o r_3) o r_4) o r_5 = r_5 o (r_3 o r_4) o (r_1 o r_2)

P_{12} -- There exist exactly three symmetry rotations
0°, 120°, 240°

The algebraic system corresponding to this embodiment, of course, can be determined by simply replacing the symmetry rotations with undefined terms. For example, we might replace the rotations by the symbols x, y, z, and the operation "followed by" by the binary operation symbol o.

Example 2: Example 2 has all of the properties of Example 1 except P_5 (the commutative property), P_{11} (which is a consequence of commutativity together with several other properties), and P_{12}. To see that P_5 does not hold, we need only observe that 120° o F ≠ F o 120°, where F corresponds to a flip about the vertical axis.

Example 3: Example 3 has all of the properties of Example 1, except P_{12}. Instead of having three elements, it has only two.

Example 4:

P_1 -- There exists at least one point.
P_2 -- Each point is on exactly two lines.
P_3 -- Each line contains exactly two points or, equivalently, there are exactly two points on each line.
P_4 -- Given any line, there exist exactly two other lines which do not have a point in common with it.
P_5 -- There exist exactly 5 points.
P_6 -- There exist exactly 5 lines.

Notice that in these statements the terms point, line, and being on, have a particular meaning assigned to them.

To form the corresponding property schemas, of course, it is only necessary to replace the meaningful terms point, line, and "on" with undefined terms such as P, L, and €. (In practice, of course, the terms point, line, and on are themselves frequently used as undefined terms.)

Example 5:

P_1 -- If the successor of a (natural) numeral, n, denoted s(n) is equal to the successor s(m) of m, then the natural numerals, m and n, are identical. (If s(n) = s(m), then n = m.)
P₂ -- The successor of any given numeral is never zero.
(For all n, s(n) ≠ 0.)

P₃ -- Any property had by zero and by the successor of any
numeral is a property of the system of numerals as a
whole.

P₄ -- The operation of addition on the numerals is commutative.
(The sum of n and m, denoted n + m may be defined as
the mᵗʰ numeral following n, or sᵐ(n).)

As in the preceding examples, the statements of Example 5 are frequently
used to describe both the properties of a particular embodiment (i.e., "the"
system of natural numerals) and the property schemas of the corresponding
system (of natural numbers). The only difference between the embodiment
and the system is whether the symbols, 0, s(0), ss(0), ..., and the successor
operation are viewed as numerals and adjoining an 's', respectively, or
are taken as undefined. (As we shall see below, it makes little difference
which way we look at this example, since the theory of natural numbers is
categorical. That is, there is no essential difference between a theory of
the embodiment and a theory of the system itself.)

Another point that should be at least mentioned concerns property P₃.
Rather than being a simple property of a system (or embodiment), P₃ is a
property of the properties of the system. Such higher order properties
cannot be formulated in the first order predicate logic directly but
require instead second order logic.

As in the preceding examples, the statements of Example 5 are frequently
used to describe both the properties of a particular embodiment (i.e., "the"
system of natural numerals) and the property schemas of the corresponding
system (of natural numbers). The only difference between the embodiment
and the system is whether the symbols, 0, s(0), ss(0), ..., and the successor
operation are viewed as numerals and adjoining an 's', respectively, or
are taken as undefined. (As we shall see below, it makes little difference
which way we look at this example, since the theory of natural numbers is
categorical. That is, there is no essential difference between a theory of
the embodiment and a theory of the system itself.)

Another point that should be at least mentioned concerns property P₃.
Rather than being a simple property of a system (or embodiment), P₃ is a
property of the properties of the system. Such higher order properties
cannot be formulated in the first order predicate logic directly but
require instead second order logic.

To say that theories deal with properties, of course, does not tell
the whole story. Something needs to be said about the logical interrela-
tionships among such properties. In particular, certain properties may
be derivable from other properties. Thus, for example, any embodiment or
system having (the commutative and associative) properties P₁, P₂, and P₅ of
Example 1 will necessarily also have property P₁₁. It would be logically
impossible for it to be otherwise. In informal theories of the sort we are
considering here, it is impossible to say more. One demonstrates that one
property is a logical consequence of certain others by arguing convincingly
that it must be true if the others are. In the next section on formal
systems, we shall see that it is possible to define logical consequence
more rigorously.

The question is, then, how does one describe an embodiment or system?
One possibility might be to simply list the properties, together with the
interrelationships between them. In certain simple cases, this might be
possible. The weakest sort of theory would consist entirely of properties
which are logically independent of one another. In this sense, a telephone
book is a physical theory about the names of people and their telephone numbers. It simply lists all of the properties of that embodiment. A physical "law," like the familiar gas law
\[ \frac{PV}{T} = \frac{P'V'}{T'} \]
of elementary physics, would in some sense constitute an even simpler physical theory, a theory consisting of a single property.

In general, of course, it would be practically impossible to list all of the properties of an embodiment or system, let alone to list the logical interrelationships between them. In most nontrivial mathematical and physical theories the properties are simply not all known.

As an alternate to simple listing, one might attempt to identify a relatively small finite list of key properties or property schemas from which all of the others can be derived as a logical consequence. In theories, such key properties are called axioms or postulates. The derivable properties are called theorems. In some simple cases, it is possible to completely characterize an embodiment or system in this way. But, unfortunately, this is not always possible. G"odel has proved a classic theorem to the effect that no matter what finite set of axioms is used, in a system as complicated as that of the ordinary arithmetic of the natural numbers, there will always be some properties which can neither be proved nor disproved.

Hence, mathematicians have been forced to actually define theories in terms of particular sets of axioms. A given set of axioms, in turn, defines a family of systems, namely, that family consisting of all systems satisfying the axioms. Furthermore, the theorems which can be derived from a given set of axioms belong to the family as a whole. These properties, axioms, are not unique to any particular system. Consider, as axioms, for example, properties \( P_1 \), \( P_2 \), \( P_3 \), and \( P_4 \) of Example 1. Collectively, these axioms define a class of mathematical objects called groups. Each of these axioms is a property (schema) of the systems (corresponding to the embodiments) of Examples 1, 2, and 3. Hence, these systems are all groups and belong to the family defined by the four axioms. A system which satisfies the defining axioms of a mathematical theory is called a model of that theory.

We note parenthetically that the term "model" is frequently not used in this sense in science. In science, "model" frequently refers to the physical theory itself or to the axioms (postulates) of that theory. While strictly speaking this is incorrect, it does not cause any real difficulty.

---

4The richer the "structure" of a theory, generally speaking, the fewer axioms are needed to define it. In "strong" theories many deep theorems can be proved on the basis of a relatively small set of axioms.
In general, when a theorist attempts to construct a scientific theory, he has a particular embodiment in mind. The theory is designed to characterize that embodiment alone. Hence, insofar as intent is concerned, the term "model" should apply equally as well to the set of axioms as to the embodiment itself. Whatever the intent, of course, this will rarely be the case. Any attempt to axiomatize a physical theory will almost invariably result in defining a family of possible embodiments.

Several other observations are worth making. First, a given theory may be axiomatized in any number of ways. When several competing axiomatizations are proposed which appear to be equally natural, a prime goal usually is to show that they are logically equivalent—that is, that each set of axioms is a logical consequence of the others. (According to Gödel's Theorem, of course, this may not always be possible.)

Second, the models of certain theories may be related to the models of other theories. Consider, for example, the theory of subgroups—in which a basic theorem is: "If S is a subgroup of a finite group G, then the order (i.e., number of elements) of group S divides the order of group G." This theorem does not strictly speaking belong to "group theory" since it is meaningful only with respect to pairs of groups. Thus, the theory of subgroups is concerned with properties of the family of pairs of groups (where one is a subgroup of the other).

Third, the conclusion of a theorem need not hold in every model of a theory. In particular, it must hold only in those models for which all of the premises of the theorem hold. For example, the conclusion of the theorem, "If the order of a group G is even, then G has an element a ≠ 0 satisfying a · a = 0" holds only in that subfamily of groups of even order. The conclusion does not hold in groups of odd order.

The preceding observations have two important consequences. One, essentially all physical theories are necessarily partial. The real world embodiments which they attempt to describe are complex beyond arithmetic and, hence, any axiomatization will necessarily be incomplete. Scientists have come to expect this, of course. No one expects a theory to be perfect and an old one is discarded only when a better one is found. New findings (even laws), which cannot be derived from existing theories (or which may even contradict them) are simply noted as exceptions. (Partial theories are central to the present development, cf. Chapters 1, and 4 through 10.)

Two, any real world display has the potential of being a model of some physical or mathematical theory. Not all displays are considered to be models, however. For one thing, a display may be so simple that no mathematician or theorist would want to even consider a theory simple enough
to have it as a model.

On the other hand, a display might be so complex that (a) no theory exists for which it is a model and/or (b) no one may yet have discovered that it is indeed a model for some existing theory. In either case, it would not be considered to be a model. Both of these possibilities are very real. As an example of the first type of situation, consider the "displays" provided by people playing games like checkers, tic-tac-toe, and so on. Clearly, such displays are extremely complex and it was not until von Neumann developed the theory of games that they could be considered to be models. As for the second type of situation, mathematicians and other theorists are continually finding new models of existing mathematical theories. Insofar as I know, our use of behavior potential to model categories and functors provides an example which is close at hand (cf. Chapter 2).

When all of the systems in a family defined by a set of axioms are isomorphic, the family is said to be categorical. In effect, a categorical theory defines a family containing a single system. This is precisely what one aims for but almost never achieves in attempting to axiomatize a physical theory. The Peano Postulates for the arithmetic of natural numbers provide a prime example. Not only do the axioms summarize much (but not necessarily all) of the key properties of arithmetic, but any system which satisfies these properties is identical up to isomorphism to the system of natural numbers. A major goal of axiomatic mathematics is to devise categorical theories. The longevity of the Peano Postulates is due in no small part to the fact that they define a categorical theory.

Of course, not all theories are categorical. Consider Examples 1, 2, and 3. They are all groups (i.e., each satisfies all of the properties required of a system to be a group), but the systems are not isomorphic to one another. Their basic sets cannot be put into one-to-one correspondence: Example 1 has order 3 (i.e., has three elements); Example 2 has order 6; and Example 3 has order 2.

Nonetheless, it is sometimes possible to prove that there is some distinguished sub-collection of systems which completely characterizes the theory in the sense that every system in the family is isomorphic to some system in the distinguished sub-collection. The theory of finite groups (i.e., groups with a finite number of elements) provides a simple example of this. It has been proved that every finite group of order n is isomorphic to some group of permutations defined on its elements (cf. Hall, 1959, 9-10). This type of result (i.e., property) is called a representation theorem and proving one effectively closes the subject to the extent that the sub-collection is understood.
In order to define formal system, we first introduce the notions of recursive set and recursively enumerable set.

**Definition 1:** A subset (of some universe) may be said to be recursive if and only if there is a mechanical procedure or algorithm for determining whether or not a given element from the universe is or is not a member of the subset.

**Definition 2:** A set is recursively enumerable if there exists a mechanical procedure for generating every element in the set.\(^5\)

A formal system may be defined as an ordered 4-tuple \((A, B, \alpha, P)\) where \(A\) is a recursively enumerable set consisting of the basic alphabet of symbols; \(B\) is a recursive subset of the expressions which can be constructed using only elements in \(A\), namely that subset which includes all and only those expressions which form meaningful entities (e.g., words, well-formed formulas (wffs), etc.) in the language in question; \(\alpha\) is a recursive subset of \(B\) whose elements are called axioms; and \(P\) is a finite set of at least binary recursive predicates, called productions or rules of inference, which are defined on the expressions of \(B\). As we shall see, the productions in \(P\) correspond to logical rules of inference; the expressions in \(B\), that can be generated by applying these rules of inference to the axioms and the expressions derived from them, are called theorems. The set of theorems is a subset of \(B\). The finite sequence of expressions (each of which is itself a theorem) produced in the process of generating a theorem is called a proof of the theorem.

Formal systems have a number of important characteristics which stem directly from the recursive nature of \(A\), \(B\), \(\alpha\), and \(P\) (a finite set is certainly recursive).

**Theorem 1:** The set of proofs generated by a formal system is recursive.

That is, given any sequence of expressions in \(B\), one can devise a mechanical

\(^5\)The notions of recursive set and recursively enumerable set are usually defined extensionally in terms of the functions which characterize them. Thus, a set is said to be recursive if its characteristic function is recursive in the sense of Chapter 2. The characteristic function of a set is that function which maps members of the set onto one and nonmembers onto zero.

A recursively enumerable set is the range of some recursive function, for example, the function which maps the set of natural numbers onto the recursively enumerable set.

According to Church's thesis, recursive (partial) functions are precisely those which can be computed by algorithm. Hence, the definitions given here are essentially the same as those given in Chapter 2.
procedure for testing to determine whether or not the sequence is a proof. The theorem follows directly from the fact that one can systematically check each step in a proof to determine whether or not it follows directly from the axioms and/or rules of inference.

**Theorem 2:** The set of theorems of each formal system is recursively enumerable.

What this means is that every theorem of the system can be generated in a finite number of steps by some procedure whose rules are all productions. All such procedures can be generated by combining the productions systematically in all possible ways. Thus, if an expression is a theorem, the system will eventually generate it and, in the process, generate a proof of it. If, on the other hand, an expression is not a theorem, the process will not terminate and one will never know whether or not the expression is a theorem.

In certain cases, the set of non-theorems of one formal system \( F = \langle A, B, \mathcal{L}, P \rangle \) may correspond precisely to the set of theorems of another formal system \( F' = \langle A, B', \mathcal{L}', P \rangle \). (Note: Both \( F \) and \( F' \) involve the same set of expressions, \( B \).) The statement logic is one such system (e.g., see Lightstone, 1964; Kleene, 1967). In this case, there is a mechanical procedure for generating non-theorems as well as theorems. This implies that the set of theorems of the original system is recursive. That is, one can determine by mechanical means whether or not a given expression is a theorem. This result follows directly from

**Theorem 3:** If both a set and its complement are recursively enumerable, then the set is recursive.

In general, of course, this will not be possible. As Church showed in 1936

**Theorem 4:** No procedure will ever be found for distinguishing between theorems and non-theorems of any formal theory as complex as number theory.

As automata logicians say, the decision problem for the set of theorems of such a system is recursively unsolvable.

Finally, we note

**Theorem 5:** For every formal system, \( F \), there exists another system \( \widehat{F} \) with a finite set of atomic symbols (in the alphabet \( \mathcal{A} \) of \( F \)) and a finite set of axioms or axiom schemas which yields (except for notation) exactly those theorems of \( F \).

This theorem provides justification for the common practice of specifying formal systems in terms of finite alphabets and finite sets of axioms.
Mathematical Preliminaries

and/or axiom schemas. (Recall that an axiom schema is an expression which indicates the form that an indeterminate number of axioms might take. For example, the axiom schema $xBx$ (where the basic alphabet is $\{A, B\}$) indicates the form taken by the class of axioms $\{B, ABA, BBB, AABAA, ABBAB, \ldots\}$.)

The tasks described by Rosenbloom (see Volume II) provide excellent examples of formal systems. They are easy to comprehend and, yet, have all of the essential ingredients. Thus, in "mathematical golf," we might have the following:

$A = \{A, B\}$

$\mathcal{B} =$ The set of expressions in the language under consideration

$\mathcal{L} = \{ABA\}$

$P = \{xBx \rightarrow AxBxA; xBy \rightarrow xByx\}$

By applying the two productions to the single axiom and the resulting theorems, one can generate an infinite but countable class of theorems. Furthermore, all of the theorems are of the form $A^m B A^n$, where $m$ divides $n$.

Generally speaking, formal systems are constructed to parallel axiomatic theories more typically studied by mathematicians. Such systems necessarily characterize relevant aspects of both the informal theory it is designed to parallel and the underlying logic necessary to justify the theorems of the theory.

As Corcoran has pointed out in Volume II, there are basically two different types of logical system, which he calls linear and suppositional. Logicians have given most of their attention to linear theories and, more particularly, to the first-order predicate logic. Without going into the details of how the basic elements of the language are formed—they are called well-formed formulas (wffs), the first-order predicate logic can be characterized as follows: Let the basic set $A$ consist of symbols for all of the elements, variables, predicates, and logical symbols that are to be used. $\mathcal{B}$, then, consists of the set of all wffs that can be constructed from this alphabet (e.g., see Lightstone, 1964, pp. 149-151). The set of axioms consists of the following five axiom schemas:

1. $A \lor A \rightarrow A$ (e.g., if $x = 5$ or $x = 5$, then $x = 5$)
2. $A \rightarrow A \lor B$ (e.g., if $x = 5$, then $x = 5$ or $x = 7$)
3. $A \lor B \rightarrow B \lor A$ (e.g., if $x = 5$ or $x = 7$, then $x = 7$ or $x = 5$)
4. $(A \rightarrow B) \rightarrow (C \lor A \rightarrow C \lor B)$ (e.g., if it is the case that if $x > 3$, then $x > 2$; then it is the case that if $x = 5$ or $x > 3$, then $x = 5$ or $x > 2$)
5. $\forall x, A(x) \rightarrow A(a)$ (e.g., if for all $x$, $x = 2x/2$, then $3 = (2 \cdot 3)/2$)
There are two rules of inference

1. \( D; D \vdash E \) (e.g., \( x = 5 \); If \( x = 5 \), then \( x^2 = 25 \).
   \( \therefore E \) Therefore, \( x^2 = 25 \)

2. \( A \supset B \)
   \( \therefore A \supset \forall t B(t) \) (e.g., If \( x = 5 \), then \( x^2 \cdot t = x \cdot t \cdot x \).
   Therefore, If \( x = 5 \), then for all \( t \),
   \( x^2 \cdot t = x \cdot t \cdot x \)

If we add to this system the axioms of a given informal axiomatic theory (stated so that they can be formalized in the first-order predicate calculus), then we have a formal theory which parallels that informal axiomatic theory.

In formalized group theory, for example, we need only to enlarge our alphabet appropriately, introduce the notion of equality, and add the following four axioms:

\[
\forall x, y \exists z, \quad x \circ y = z \\
\forall x, y, z, \quad (x \circ y) \circ z = x \circ (y \circ z) \\
\exists 0 \forall x, \quad x \circ 0 = x \\
\forall x \exists x', \quad x \circ x' = 0
\]

Since linear systems of logic (i.e., theories of proof) are logically equivalent to suppositional theories and vice versa, one could in principle reformulate a system like that given above along the lines suggested by Corcoran. Although the standard approach taken in formalizing axiomatic theories involves linear systems of logic, I am convinced by Corcoran's arguments that suppositional theories of proof are more natural from a psychological point of view. More is said about this in Chapters 4 and 6.

5. PROOF THEORY AND MODEL THEORY

Meta-mathematics or proof theory is the informal study of formal mathematical systems. As Kleene (1967, 199) puts it, an informal proof in meta-mathematics is a proof of a meaningful statement about meaningless formal objects. The proof theorist, however, is very restricted in the kinds of (informal) proof which he is allowed to use. In particular, he may only use finitary methods, no appeal to the infinite is allowed.

Meta-mathematical theorems tell us such things as whether or not a

---

6 The Peano Postulates of elementary number theory do not satisfy this condition. They require second-order logic for their formulation.

7 Note that these are axioms and not axiom schemas.

8 Presumably, linear theories are preferred by logicians because they are easier to work with.
formal proof exists in a given (formal) system without exhibiting such a proof, or whether or not a particular formal system is consistent (i.e., free from contradiction). Among the more central theorems of proof theory are the following

**Theorem 1** (The compactness theorem): A set of sentences is consistent if each nonempty, finite subset of sentences is consistent.

**Theorem 2** (Church's theorem): No algorithmic procedure exists for determining whether or not an arbitrary wff of elementary number theory is a theorem.

Model theory complements proof theory in the sense that it deals with relationships between algebraic systems and families of such systems, on the one hand, and the properties of sentences or sets of sentences in formal theories (i.e., systems), on the other hand. A major concern has been the relationships between properties holding in algebraic systems and deducibility of statements within formal theories which parallel these algebraic systems.

While the ultimate objective of model theory is to prove things about the relationships between formal theories and algebraic systems (i.e., models of these formal theories), model theorists typically do not work directly with algebraic systems but with what are called structures. This approach simplifies technical details considerably with no loss of mathematical generality because it is possible to construct an algebraic structure which corresponds to any given algebraic system. We might note, parenthetically, that in proving theorems in model theory some appeal to the infinite is necessary and allowed.

For some readers, it may be helpful to consider briefly what structures are and, in particular, to consider just how formal theories are related to structures and how structures, in turn, are related to algebraic systems. The reader who is not interested in details may move directly to Theorem 3.

Suppose we are given a formal theory, $F$, and wish to construct a structure, $M$, which can be described by the axioms and theorems of $F$. The basic idea is to set up a one-to-one correspondence between the constant and relational symbols of $F$ and the distinguished elements and relations of $M$. Then, the wffs in $F$ are said to be defined in $M$ if the object and relation symbols in the wff all correspond to distinguished elements and relations of $M$. The variables in a wff are presumed to range over the elements in the basic set of $M$.

When each variable in an atomic wff of $F$ (i.e., a relational symbol) is replaced by one of its values, the corresponding relation in $M$, either holds or does not hold for the corresponding n-tuple of $M$. (The elements in this n-tuple of $M$ correspond to the values of the constants in the relation of $F$.) An atomic wff is said to hold if the corresponding relation holds at the indicated values. More complex wffs are said to hold according to certain requirements on the atomic wffs of which it is composed. For example, $R(a, b) \lor Q(c)$, (where $\lor$ means "or") would be said to hold if at least one of the two relations in $M$, corresponding to $R$ and $Q$, holds at the n-tuple of elements corresponding to $a$, $b$, and $c$, respectively. (In
doing research in the subject, the object and relation symbols in F and
the corresponding elements and relations in M are often viewed as identical.
This simplifies notation. For more details, the reader is referred to
Robinson (1965) and Lightstone (1964).)

With this background, we can make the notion of a model more precise.

**Definition:** If each wff in a set of wffs, K, holds in a structure, M,
under a one-to-one correspondence, then M is said to be a **model** of K.

The nature of the relationship between algebraic systems and structures
can be seen as follows. Let $\mathcal{J}$ be an algebraic system. Change all of the
operations (functions) of $\mathcal{J}$ into equivalent relations. Next, consider the
class of all embodiments of $\mathcal{J}$ and form a new entity $\mathcal{J}'$ ($\mathcal{J}'$ is not a system)
such that:

1. The basic set is the union of all sets of basic elements of the
   embodiments.
2. The relations and distinguished elements of $\mathcal{J}'$, respectively, are
   sets of the corresponding relations and distinguished elements of
   the embodiments of $\mathcal{J}$.
3. Introduce an equivalence relation on the elements in the basic set
   (of $\mathcal{J}'$) which partitions the basic set into equivalence classes
   corresponding to the original elements of $\mathcal{J}$.

The introduction of the equivalence relation ensures that if an n-tuple, $n_1$
satisfies a given relation in one of the sets of relations and another
n-tuple, $n_2$, is element-wise equivalent to $n_1$, then $n_2$ also satisfies the
relation. $\mathcal{J}'$ defined in this way is called an **algebraic structure**. (In
fact, model theory typically deals with arbitrary structures and does not
require the introduction of an equivalence relation.)

The particular relations in the sets of relations of algebraic struc-
tures correspond to relations of the embodiments of the original system $\mathcal{J}$.
Hence, the defined equivalence relation allows any given relation (in a
set of relations) to be satisfied by an n-tuple of elements of an arbitrary
embodiment. Indeed, n-tuples involving elements from up to n different
embodiments may satisfy the relation. In this sense, the definition of an
algebraic structure does not distinguish between the relations in a given
set of relations except by name—even though the relations correspond to
different embodiments.

The following theorems are basic in model theory. In 1948, Henkin
proved

**Theorem 3:** (The Extended Completeness Theorem): A theorem
can be deduced from a set of axioms within the lower predicate
logic if and only if the theorem holds in every model of the
set of axioms.

This amazing theorem helped to pull together a number of basic completeness
results going back to Gödel's completeness theorem of 1930.

**Theorem 4:** A theorem can be deduced if and only if it is a
tautology (i.e., if it is true).

Another consequence of the extended completeness theorem is a theorem
proved earlier that is known as the Skolem-Lowenheim Theorem.

**Theorem 5:** If $K$ is a set of wffs that possesses a model,
then there is a model of $K$ whose basic set is denumerable
(i.e., finite or countably infinite).

Finally, we note that it is not generally possible to formally charac-
terize an arbitrarily given system although this was originally thought to
be possible by Hilbert and his followers. In what is perhaps the most
amazing result of modern logic, Gödel (1931) proved (roughly)

**Theorem 6:** No one can hope to ever devise a consistent
formal theory, which is complete, for a system as complex
as number theory.

What this "incompleteness" theorem says, in effect, is that given any
consistent formal theory (or axiomatic equivalent) of number theory, there
will necessarily be statements about number theory which can not be proved
nor disproved within this theory. Some mathematicians believe that Fermat's
famous "last theorem" is of this type. That is, they feel that the theorem
can neither be proved nor disproved within standard formulations of number
theory.
Chapter 4

RULE-BASED THEORIES OF STRUCTURED KNOWLEDGE: GENERAL NATURE AND RELATIONSHIPS TO LINGUISTICS

This chapter is concerned with the general problem of characterizing structured knowledge (competence) in terms of rules. In Section 1, we introduce the notion of a rule set and identify the major criteria involved in evaluating different rule sets. Then, we consider some alternative ways of characterizing structured knowledge in terms of rule sets. Because of its clarity of structure, mathematics provides a convenient source for illustration. In section 3, some relationships between present goals and those of linguistics are discussed. A formalization and extension of the theory is given in Chapter 5 which at once is more rigorous and more general (e.g., it deals with perception and semantics). The mathematically inclined reader may want to begin there, realizing all the while that the promised rigor is relative to existent behavioral theories.

1. RULE SETS AND EVALUATIVE CRITERIA

The goal of any rule-based theory of knowledge is to provide an account of a given class of input-output pairs (behavior class). (Behavior classes may be thought of as that portion of a subject's potential behavior that is of interest to an observer.) Specifically, the goal is to identify a finite set of rules that makes it possible to generate each and every output from some input in the given behavior class.

There is, unfortunately, no completely systematic way to go about
identifying rule sets. About all that can be said is that rule sets are necessarily based on a limited amount of data (from a behavior class). In attempting to devise a grammar (i.e., rule set) for English, for example, the theorist draws on his experience with the language in order to identify a rule set which generates all and only sentences of English. This experience is necessarily limited. The theorist does not introduce a transformational rule between active and passive forms of a sentence because he has had direct experience with all such sentences, for example. Rather, he does so because he knows from experience that this relationship is important and comes up over and over again.

Needless to say, except for very simple behavior classes (e.g., where the number of input-output pairs is finite), devising an adequate rule set can be a very demanding task. Indeed, some behavior classes are so complex that no one has succeeded in coming up with a completely adequate rule set. Our present inability to account for (the generation of) any natural language provides the standard example.

The problem of how to evaluate alternative characterizations, then, is of major importance. As with any theory—in this case a theory about the knowledge had by given individuals, a primary requisite is power or generality. All other things being equal, one characterization is to be preferred to another to the extent that it accounts for more different kinds of behavior (in the given class).

The phrase "more different kinds of behavior," of course, is ambiguous and, in general, depends ultimately on the theorist's intuition. Although use of such criteria may sound antithetical to science, a good case may be made for allowing them, especially at the present rudimentary state of scientific knowledge in the field (e.g., see Chomsky, 1965, pp. 18-27). Nonetheless, in some cases, it may be possible to specify the criteria more rigorously. If the given class happened to pertain to division problems, for example, then it would be a relatively simple matter to identify various subtypes (e.g., one-digit divisor, n-digit divisor less than first n-digits of dividend).

In addition to the power of a given characterization, there are also considerations of simplicity and internal consistency—in a word, parsimony. Clearly, a major concern in evaluating alternative rule sets is the number of rules involved in each, but parsimony involves other considerations as well. Among these considerations is what might be called "intrinsic simplicity or cohesiveness." By this is meant such difficult-to-operation-alize notions as the degree to which the rules conform to intuition—the extent to which the characterization seems natural as opposed to having a
post hoc or arbitrary character.2

A theory of knowledge cannot be judged solely in terms of the power and parsimony of its rule set, however. If it is to have behavioral significance, the theory (of knowledge) must also be compatible with some theory of performance which tells precisely how that knowledge is to be used.

More specifically, the way the rules in rule sets may be combined in accounting for input-output pairs must be compatible with the mechanisms which govern human performance. Without this requirement there would be any number of equally plausible modes of interaction. One possibility would be to require that rules be applied in a discrete manner so that the only input-output pairs that can be generated are those associated directly with some rule (in the rule set). Another possibility would be to allow rules to be applied in succession (i.e., to allow composites of the rules in rule sets). Still a third possibility might be to allow rules to operate on other rules, thereby generating new rules which in turn can be used to account for still more diverse input-output pairs.

The behavior classes which can be generated by any given rule set clearly will depend on how the rules can be combined (if at all). It is these laws of combination which must be compatible with some behavior theory. This will insure that the theory of knowledge can be embedded in the more encompassing behavior theory. In the next section, we shall consider the above laws of combination in more detail, and will select one as most promising. Throughout Chapters 4, 5, and 6, we simply assume that adequate performance mechanisms are available to the subject. This debt is discharged in Chapter 7.

Unlike power and parsimony, this type of "relative" criterion has not to my knowledge played an important part in evaluating competence theories. Its sole purpose is to insure behavioral relevance.3

---

2Precision and reliability of theories and the degree of control they permit over the environment are also frequently mentioned bases for evaluating theories (e.g., Meehan, 1968). As should become increasingly clear, each type of theory we shall propose ranks high on each of these criteria. Special mention is made only here this provides a useful basis for comparison. Degree of control, for example, is discussed at some length in Chapter 8, which deals with the mechanisms of motivation.

3Explicit attention to similar criteria could lead to a deeper understanding of the role and relative importance of such diverse notions as natural deductive systems in logic, and transformational grammars in linguistics. For example, if one is interested solely in providing a rigorous and valid account of logical reasoning, then there is no need to get involved in natural deductive systems of the sort Corcoran has discussed in Volume II (also see Chapter 6). (continued on following page)
A closely related consideration has to do with the extent to which a given rule set seems to capture what any particular individual knows, independently of the behavior class in question. If the rule set is to account for the performance of an individual, then it is necessary but not sufficient that the laws of interaction be consistent with the general behavior mechanisms. In addition, the rule set must be consistent, in a sense made explicit in Chapter 7, with what the subject knows. For present purposes it is sufficient to note that the relevant knowledge had by the subject must be representable in terms of the rules in the rule set. The basis upon which the theorist identifies such rule sets is, presumably, the common culture he shares with the subject and/or his previous experiences with the subject (or others like him).

2. ALTERNATIVE CHARACTERIZATIONS

In this section we consider some alternative ways of characterizing structured knowledge, and evaluate each with respect to power, parsimony, and kind of performance assumptions necessary for psychological viability.

Undoubtedly, the simplest way to account for a finite class of behaviors is just to list the behaviors. Thus, for example, a list of paired associates can be characterized as a finite set of discrete associations (i.e., degenerate rules). Lists of paired associates, of course, are not what we usually have in mind in talking about complex behavior, and, in any case, characterizations which consist of simple lists of associations would be essentially sterile in content. Even simple addition requires an infinite number of input-output pairs.

A somewhat more realistic way to account for a class of behaviors derives from recent attempts in educational circles to define school curricula in terms of a finite number of operational objectives (e.g., Lipson, 1967). Each objective of such curricula defines a class of rule-governed behaviors; the abilities to add, to multiply, to find areas of triangles, and so on, provide obvious examples. Each such curriculum, then, can be characterized by the union of a finite set of rule-governed classes. In turn, this union can be accounted for in terms of a rule set consisting of

\[ \text{(continued)} \]

Natural deductive systems add nothing toward obtaining new results in logic and, in fact, have been shown to be logically equivalent to the simpler linear systems with which most logicians work. There may be many reasons for preferring natural deductive systems, on the other hand, where one is concerned either with characterizing the kinds of proofs mathematicians actually write (Corcoran, Volume II) or with eventual compatibility with behavioral science. Apparently, natural deductive systems more closely parallel the kinds of proofs that appear in mathematics texts, and in turn, the way people actually make deductions.
one rule for each (rule-governed) class, under the assumption that these rules act independently of one another. Although most curriculum constructors who have followed this approach have stopped short of identifying the underlying rules, it is clear that this can be done (cf. Scandura, Durnin, Ehrenpreis, & Luger, 1971). (For further discussion of this and related issues, see Scandura (1970b, 1972b) and Ehrenpreis & Scandura (1972).) 4

In general, the list type of characterization has the major advantage of requiring a very simple performance mechanism. If knowledge is characterized as a list of discrete rules, a theory of performance need only tell how and when individual rules are put to use. Since the rules act in a discrete manner, no interaction among them need be postulated.

This advantage, however, is also its major disadvantage. Because the characterizing rules are discrete, they cannot account for behaviors that are not directly associated with some rule (in the rule set). That is, a given rule set can account only for those behaviors which can be generated by direct application of one of the characterizing rules. Suppose the rule set only included rules, say, for adding, subtracting, multiplying, and dividing. In this case, for example, the subject would be unable to generate the subtraction facts corresponding to a given addition fact.

One might counter, of course, that it would be a small thing simply to add the relational rules

\[ (1) \quad a + b = c \rightarrow c - a = b \]
\[ a + b = c \rightarrow c - b = a \]

to the original list. Indeed, this is precisely the sort of reply given by curriculum constructors of the operational objectives persuasion when confronted with the criticism that their objectives do not constitute a mathematically (or otherwise) viable curriculum. (Lists which include relational rules are referred to as relational lists.)

The trouble with this sort of argument is that it misses the point entirely. Not only would such an approach be ad hoc—which really says nothing in itself except to convey some ill-defined dissatisfaction, but it would be completely infeasible where one is striving for completeness. To see this, it is sufficient to note that a new rule would have to be introduced for every conceivable interrelationship, and that the number of such interrelationships is indefinitely large. One could easily envision a number of rules so large that no human being could possibly learn all of them in a single lifetime. The sum total of all mathematical knowledge

---

4We note parenthetically that since the number of rules in any characterization is necessarily finite, they can be replaced by a single procedure in which each of the original rules corresponds to a (possibly different) path.
which is presently in print, for example, is so vast that no one has, or could possibly, acquire all of it. As vast as this knowledge is, however, a really good mathematician is capable of generating any amount of new mathematics which does not appear anywhere in print. That is, he can create. Much of the new mathematics might be utterly trivial, of course, but the very fact that it exists at all strongly suggests that any characterization such as that described above would almost certainly miss much that is important.

Limiting a subject's performance capability to simple application of (discrete) rules would not allow for the routine diversity of human behavior. Suppose, for example, that a subject knows how to add, and that he knows rule (1), which relates addition and subtraction, but that he does not know explicitly how to subtract. Under these circumstances, one might still reasonably expect (at least certain) subjects to perform satisfactorily on subtraction problems (cf. Chapter 6). But, given only the simple performance capability assumed above, a relational list of this sort would not suffice to account for such behavior.

The power of a relational list could be increased, however, by allowing for the combination of simple and relational rules. Thus, relational rules in a characterizing rule set might be allowed to operate on (input-output) instances of other rules in the rule set, thereby generating new instances. These new instances, in turn, would make it possible to generate new responses (i.e., outputs). Relational rules (1), for example, might operate on instances of a known addition rule (i.e., addition facts) and generate corresponding subtraction facts.

There are some important limitations of even this type of relational characterization, however. First of all, the approach lacks both power and parsimony when extended to more complex domains of behavior. To see this, note that relational rules (1) are but one of a class of similar relational rules between binary operators and their inverses. Another relational rule in this class is

\[ r_1 \oplus r_2 = r_3 \leadsto r_3 \ominus r_2 = r_1 \]

where \( r_1, r_2, \) and \( r_3 \) are rotations, \( \oplus \) is the operator "followed by" applied to rotations, and \( \ominus \) is the inverse operator, "take away." Because the number of relations in even one such class is indeterminately large, this approach, too, would lack parsimony and have an unnecessarily ad hoc flavor. Another possibility would be to generalize relational rules (1) so as to account for additional inverse rules. (Rule A is said to be more general than rule B if the set of instances of B is a proper subset of the set of instances of A.) Rules like (1), for example, can easily be (cont'd.)
As we shall see in Chapter 7, a more important limitation is that people simply do not perform in this way. Available empirical evidence provides strong support for performance mechanisms which more naturally involve higher order rules, which act on classes of (lower order) rules, rather than relational rules, which act on classes of instances of rules. (Higher order rules correspond to functions which act on and/or generate other functions.)

The inverse rule by which each lower order rule (operation) is mapped into its inverse is an example of a higher order rule. Specifically, this higher order rule corresponds to a unary function since it maps single rules into other rules. Higher order rules, of course, may operate on n-tuples of lower order rules. The higher order composition rule, for example, generates the composition of any pair of rules such that the output of the first corresponds to the input of the second. When applied to rules for converting yards into feet and feet into inches, for example, the composition rule generates a rule for converting yards into inches.

Allowing rules in rule sets to act on other rules greatly increases explanatory power and parsimony. Consider, for example, a rule set that includes: (a) an addition rule

\[ 5a, b + c \]

(continued) generalized so that they apply to any number of different kinds of operators and elements. Thus, a general scheme might be introduced for permuting arbitrary elements in a fixed way, and, say, adding a prime to each operator (e.g., the inverse of + would be written +'). This might appear to be a viable alternative until we realize that the elements (e.g., numbers, rotations) and the operators (e.g., addition, followed by) operate at different levels. In particular, each operator applies to a class of elements. Generalizing in this way would be undesirable just to the extent that this difference between operators and elements is reflected in behavior. In particular, this approach would not allow for learning instances in "chunks," according to the operators in question. Each instance of such a rule would be treated on an equal plane. Thus, for example, a subject who has mastered the instance

\[ 180° + 90° = 270° \rightarrow 270° + 90° = 180°, \]

would, according to this view, be just as likely to demonstrate future mastery on

\[ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \]

\[ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \]

as on

\[ 90° + 270° = 360° \rightarrow 360° + 270° = 90°. \]

This clearly runs counter to behavioral intuition and general observations on how people perform in such circumstances. To be sure, Scandura and Durnin (1968) have obtained data which suggests that mastery of instances like (4) is more directly tied to performance on instances like (2) than are instances like (3).
in the system \( A \) of natural numbers, and

(b) the inverse rule which maps operators into their inverses.

Clearly, this rule set accounts for subtraction. Application of the inverse rule to the addition rule generates the subtraction rule. The subtraction rule, in turn, makes it possible to generate the solution to any subtraction problem. Furthermore, for each new rule we add to the rule set, we get another—its inverse—free.

As will become increasingly clear, in this and the following chapter, higher order rule sets allow for a good deal of what normally goes under the label of creative behavior. Newly derived rules, for example, may act in turn on other rules to generate still more (new) rules. In this way, rule sets can "grow" very fast indeed, and more important here, make it possible to account for a wide variety of behavior, including behavior which may appear to be quite far removed from the rules immediately available at any given point in time. For the sake of argument, suppose that to the above rule set we add the higher order rules of composition and a rule that maps binary operations (e.g., addition) in systems involving numbers into binary operations (e.g., followed by) in isomorphic systems involving, say, rotations. For example, consider the classes of systems which contain systems of natural numbers modulo \( n \) (where \( n = 1, 2, \ldots \)) and systems of rotations of size \( k(360^\circ/n) \). This rule set, then, would account for input-output pairs as different as those corresponding to inverse operations in systems of rotations. (Chapter 6 is concerned with identifying some of the specific kinds of rules which are characteristic of mathematical knowledge and with showing how such rules may be combined to produce a wide variety of mathematical behaviors.)

Although devising higher order rule sets might appear to be more difficult than, say, discrete lists, this is not true. In fact, the basic principles are the same. In both cases, we begin with a finite corpus (set of input-output pairs) and attempt to identify rules which account not only for the corpus but for the rest of the behavior class. In the higher rules formulation, however, we do not stop with identifying rules for generating outputs from inputs. We look also for relationships among the rules themselves. This frequently makes it possible to identify higher order rules.

Although it is sometimes harder to identify higher order rules, the approach has a major advantage over discrete lists. Because higher order rules may operate on indeterminately large classes of rules, and may thereby generate any number of new rules, it is not necessary to identify all of the needed lower order rules directly. Indeed, our experience has been that a reasonable amount of insight can substantially reduce the amount of drudgery required. It is beyond the scope of this book to go into the details of this process and the interested reader is referred to Scandura (1971a, 1972b). The feasibility of the process is indicated by the fact that an entire book (Scandura, 1971a) has been subjected to such analysis and the results published as a workbook (continued)
Finally, we note that this increased power is gained without any loss of generality; in particular, the higher-order rules conception allows relational rule sets as a special case (where the number of instances is finite). This is accomplished by introducing a trivial reformulation of the relational type of characterization in which each instance is replaced by a degenerate rule (i.e., association)—hence, the necessity of keeping the number of instances finite. For example, the instance \((4+5, 9)\) (or, equivalently, \(4+5=9\)) is replaced by the rule \(4,5 \rightarrow 9\). In this case, application of relational rules \((1)\) to addition associations generates associations involving subtraction (e.g., \(9,5 \rightarrow 4\)).

3. RELATIONSHIPS TO LINGUISTICS

Following Chomsky (1957), the fundamental concern of linguistic theory is the construction and justification of grammars. A grammar of a language is essentially a theory of it. Any scientific theory is based on a finite number of observations and it seeks to relate the observed phenomena and to predict new phenomena by constructing general laws in terms of hypothetical constructs. Similarly, a linguistic grammar (say of English) is based on a finite corpus of utterances (i.e., observations), and it will contain certain grammatical rules or laws stated in terms of particular phonemes, phrases, sentences, etc., of English. (These are the hypothetical constructs.) For example, the rules might express structural relationships among the sentences of the corpus and the indefinite number of sentences generated by the grammar beyond the corpus, which constitute the predictions. The fundamental problem of linguistic theory, then, is to develop and clarify criteria for developing and evaluating grammars.

Chomsky has taken a dim view of the possibility of constructing a general (universal) linguistic theory which provides a practical and mechanical method for actually constructing a grammar, given a corpus of utterances. He has suggested instead that linguists adopt the weaker requirement that a general theory of linguistic structure provide an evaluation procedure for evaluating the relative effectiveness of different grammars.

The fundamental task of linguistic research, then, is to propose and evaluate (justify) theories of linguistic competence, just as ours are to...
evaluate theories of competence generally. Thus, the grammar of a language may be thought of as the "knowledge" had by an idealized individual which makes it possible for him to generate all and only sentences of that language, and, conversely, to correctly interpret sentences of it. (In this regard, Miller and Chomsky (1963) have argued that there is a reciprocal relationship between a theory of the speaker and a theory of the listener. An adequate grammar, according to this view, represents a theory of both.)

A minimum requirement of any grammar is that it must be based on a finite number of rules. The human mind is finite by almost any reckoning, and any theory which makes reference to an infinite number of rules can at best bear an approximate relationship to human knowledge.

Arguing in this way, it can be shown that English, for example, is not a finite-state language (e.g., Chomsky, 1957). That is, it is impossible to construct a finite-state Markov device which constructs, word by word, all and only grammatical sentences of English. The problem essentially is that embedding sentences within (other) sentences requires an infinite number of states.

A good deal of effort has also gone into exploring the relative strengths and weaknesses of generative (phrase-structure) grammars, which generate sentences and impose (surface) structures on them, called P-markers. While no one has shown that generative grammars are necessarily inadequate in this respect, Chomsky (1957) has argued that such an account of language behavior fails to consider adequately the fact that different sets of sentences may bear important relationships to one another.

To take account of these relationships, he introduced the notion of grammatical transformations between sets of related P-markers (Chomsky & Miller, 1963, 301). Transformations may be viewed as rules of the form \( \emptyset_1, \emptyset_2, \ldots, \emptyset_n \rightarrow W \) where the \( \emptyset_i \) and \( W \) may be thought of as P-markers (Chomsky & Miller, 1963, 296-306). For example, transformation \( T \) maps P-markers \( \emptyset_1 \) and \( \emptyset_2 \) into \( W \).

7In generative linguistics, the theorist is given a finite corpus, say, of sentences—and is required to come up with a characterization which accounts for these and other sentences of the language. The approach is deterministic, as opposed to stochastic, although the latter approach too has been pursued with some vigor (e.g., Miller and Chomsky, 1963). The primary emphasis has not been on just generating sentences, however, but also on imposing (generating) structures on sentences which, on the one hand, parallel their respective phonetic interpretations (surface structures) and which, on the other hand, parallel their respective semantic interpretations (deep structures) (Chomsky, 1968, 24-28). This refinement will concern us here only indirectly.
Chomsky further argues that transformational grammars, which include both transformations and phrase structure rules, provide a far more efficient account of certain aspects of natural languages. The use of transformations, for example, makes it possible to account for such things as tense, formation of compound sentences, and certain ambiguities in sentence structure, which can only be dealt with in a post hoc fashion by phrase-structure grammars. More generally, the use of transformations appears to be essential in assigning deep structures to sentences in a way which reflects their underlying meaning.

Although the goals of linguistic and other structural theories of competence are very similar, there are important differences, especially as the latter are developed here. These differences include both the kinds of rules proposed and the criteria used to evaluate the theories. In general, many of these differences are ones of emphasis but some reside in the fact that with a few notable exceptions (e.g., Miller & Chomsky, 1963, 483-488), linguists have too infrequently considered the relevance of their work to behavioral science.

We begin with the observation that many linguistic rules of competence are not directly reflected in the kinds of behavior generally considered by psychologists (in studying language behavior). Thus, for example, transformational rules allow for the transformation of one set of structures into another. It is unlikely, however, that anyone but a budding linguist would be asked to demonstrate this sort of behavior directly. This is not true of mathematics, however. Knowledge of relationships—indeed, of second, third, and even higher order relationships—lies at the very core of mathematics.

As an example of an ambiguous sentence consider, "Flying planes may be dangerous," which has two distinct meanings (deep structures):

1. (Flying planes) may be dangerous.
2. (Flying) planes may be dangerous.
of any reasonable interpretation of knowing mathematics. Furthermore, such behaviors are frequently tested directly. This is probably as much a matter of tradition, however, as anything else, as there is no fundamental reason why psycho-linguists could not test for transformational competencies directly.

Furthermore, although the role transformations play in linguistics is similar to the role higher order rules play in our theory, they are not identical. Transformational rules operate on and generate P-markers of sentences. P-markers, in turn, may be thought of as instances of rules which generate (surface) structures of sentences. Hence, transformational rules may be thought of as mapping instances of (generative) rules into other rules. In this sense, transformations are like relational rules which map instances of rules, like addition, into instances of another, like subtraction. For example, the relational rule

\[ a + b = c \rightarrow c - b = a \]

where \( a, b, \) and \( c \) are numbers, is of essentially the same form as transformational rules. In view of the close relationship between relational rules and transformations, the latter may also be viewed as degenerate forms of (higher order) rules which act on associations. (See the discussion above concerning the relationship between relational and higher order rules.)

In effect, allowing rules to act on other rules is a considerably more general notion than allowing transformations to act on P-markers. There is no direct counterpart for the former notion in present-day linguistics. In linguistics, the most general case (of higher order rule) would seem to call, for example, for rules which act on classes of transformations in different languages. Composition might be a good candidate. Other such rules might involve reference to meaning, the common denominator for all languages.

Linguistic theories of competence in the Chomsky tradition also tend to be restrictive in another sense. According to Chomsky (1968, 23)

"A person who has acquired knowledge of a language has internalized a system of rules that relate sound and meaning in a particular way. The linguist constructing a grammar of a language is in effect proposing a hypothesis concerning this internalized system."

---

9Actually, these are procedures composed of rules in a generative grammar.

10The associations on which transformational rules act are instances of procedures which have certain phrase structure rules in common.

11So-called "universal grammars" may also be involved in some way but as the discussion below indicates, they appear to have more in common with learning and performance mechanisms.
Thus, the major reason for constructing a grammar is to account for the process by which an idealized subject is able to interpret statements in a language—or, conversely, to express ideas in a given language. The competencies (rules) necessary for relating different sounds, for example, seem not to have been dealt with in a serious way within Chomskian linguistics. Much the same may be said about rules which operate at a purely semantic level—although universal grammars$^{12}$ may well be involved here. In contradistinction, an account of structured knowledge may be as concerned with rules which operate entirely within levels as between them.

There are also important differences in the way in which linguistic and structural theories of competence may be evaluated. First, consider power and parsimony. In linguistics, the traditional approach has been to evaluate grammars by determining the extent to which they are compatible with linguistic intuition. If the sentences and structures (on the sentences) produced by a grammar are in accord with what a mature speaker or hearer feels they should be, the grammar is said to be a good (or feasible) one. To the extent that they are not in accord, the grammar is said to be inadequate.

Our own view is somewhat different. Rather than attempt to evaluate all characterizations with respect to a fixed criterion, we prefer to allow for more diversity. This approach seems somewhat more appropriate because of our interest in characterizing the knowledge had by different individuals, naive ones as well as mature.

Such an approach would also be more useful in curriculum construction where the emphasis is on the knowledge associated with different curricula. Accordingly, it may sometimes turn out that two or more characterizations are equivalent with respect to one class of behaviors (used to evaluate them) but not another. Thus, for example, in curriculum construction, a list of the four computational algorithms and a more "meaningful" characterization of arithmetic may both account equally well for the behavior associated with arithmetic computation, but differ greatly in other respects.

We do not, of course, hold out any real hope of ever characterizing structured knowledge in a complete sense. It is almost impossible to conceive of what it would mean, for example, to know all there is to know about mathematics. This obviously is also true of language, and recently more emphasis is being given to accounting for the behavior of different classes of individuals, such as young children, the underprivileged, or

$^{12}$See the discussion below.
Linguistic and structural theories of competence may both be evaluated in terms of their compatibility with more fundamental criteria, although here again there may be important differences of emphasis. Quoting from Chomsky (1968, 24)

"The principles that determine the form of grammar and that select the grammar of an appropriate form on the basis of certain data constitute a subject that might ... be termed 'universal grammar.' The study of universal grammar ... is a study of the nature of human intellectual capacities. It tries to formulate the necessary and sufficient conditions that a system must meet to qualify as a potential human language, conditions that are...rooted in the human 'language capacity,' and thus constitute the innate organization that determines what counts as linguistic experience and what knowledge of language arises on the basis of this experience. Universal grammar, then, constitutes an explanatory theory of a much deeper sort than particular grammar, although the particular grammar of a language can also be regarded as an explanatory theory."

Among the principles of universal grammar cited by Chomsky (1968) are the principle of cyclic application of phonological rules and the A-over-A principle. From Chomsky's quotation, it seems clear that the role of universal grammar is in many ways analogous to the role promised for the learning and performance mechanisms of Chapter 7. In each case, the theory of knowledge (competence) proposed must be compatible with a deeper set of principles.

The emphases, however, are not the same. In particular, principles of universal grammar have been formulated to account for problems relating specifically to linguistic competence rather than for learning and performance more generally. Our own formulation is designed to have broader relevance—to the learning and performance associated with all kinds of structures. Specifically, we have argued that rules should be allowed to operate on rules, and that this assumption is reflected in mechanisms which actually govern human behavior. To the extent that this proves to be correct (cf. Chapters 7 thru 9), one might expect that particular principles of universal grammar may turn out to be special cases of ours. The universal principle of cyclic application, for example, appears in some ways to be similar to our rule for forming composites. Any attempt to demonstrate such a relationship is beyond the scope of this development, however, and is an important task for future research.
Following the traditional distinction in axiomatics between theories (descriptions of systems) and systems themselves, there are two basic kinds of behaviorally relevant theories of knowledge. The former type is descriptive and is well illustrated by Piaget's description of developmental stages.* Both he and some of his followers have attempted to describe certain essential features of behavior. The goal, then, is to taxonomize and characterize such descriptions (of behavior). The present approach falls in the latter category. Here the goal is not to describe behavior but to show how it may be generated.*

The goal of any such theory of knowledge is to provide an account of a given body of data. In the present theory, we are given a class of stimulus–response pairs and our job is to provide specifications for how this class of pairs may be generated.

The purpose of this section is: (1) to specify the nature of the stimuli (inputs) and responses (outputs) to which the theory relates, (2) to define precisely what is meant by rules and their extensions, (3) to identify some of the kinds of higher order rules allowed in the theory, (4) to specify the form of the theory itself (including the laws governing the interaction among rules), (5) to illustrate the theory, and (6) to suggest a general axiomatization and to propose some conjectures.

The theory so defined provides a schema to which specific realizations of the theory must conform. Any particular realization of the theory involves a finite set of rules which, given the laws governing their interaction, accounts for the behavior of interest. A specific rule set, together with the laws of interaction, constitutes a theory of the given behavior. In this sense, a theory of knowledge plays the same role as does a grammar of a language (i.e., the grammar is a theory of the language, *See footnotes 1 and 2 at end of chapter.*
1. NATURE OF THE STIMULI AND RESPONSES

Among the undefined terms of the theory of knowledge are elements (E) and the relation of being an input element for an output element (cf. Hocutt, 1967). Since the theory is designed to have behavioral relevance, we single out those input-output pairs that are potentially observable and refer to them as stimulus-response (S-R) instances (or, equivalently, (S,R) pairs). The class of S-R instances is a subset of the Cartesian product set \( \mathcal{S} \times \mathcal{R} \). We denote this subset by \( \mathcal{S} : \mathcal{R} \). The terms "input" and "output" are used more broadly to refer to arbitrary inputs and outputs of rules (whether they are stimuli and responses or not).

Although these S-R instances correspond to potentially observable events in the real world, we are not concerned in the theory of knowledge with how the S-R instances in \( \mathcal{S} : \mathcal{R} \) are determined. They are simply given. (Equivalently, the contents of \( \mathcal{S} : \mathcal{R} \) depend on the interests of the theorist.)\(^3\)

The given set \( \mathcal{S} : \mathcal{R} \) of S-R instances, then, could form the starting point of our development, and for many purposes this is the most convenient thing to do. For other purposes, however, especially where we are concerned with perceptual and decoding phenomena, this is not sufficient. In this case it is not sufficient to simply identify the S's and R's of interest. It is also important to specify the respective structures of these stimuli and responses.

To accomplish this, we begin with a set of unanalyzable (atomic) stimuli and responses and let the inputs and outputs of interest be complexes (or arrays) of the atomic stimuli and responses. Examples of atomic stimuli and responses are provided by the black and white dots from which images

---

\(^3\)Presumably, S-R instances are determined on the basis of some set of criteria known to the theorist and of interest but which for one reason or another (e.g., there may be too many of them to list) he has never made public. The reason that the theorist may be afforded this degree of freedom is the common culture he shares with other scientists. The linguist, for example, may take phonemes, morphemes, words, sentences, or the paragraph as his basic unit of analysis depending on which aspects of language concern him most. The structural learning theorist has even more latitude.

Put differently, the theorist always keys on certain properties of the stimulation he receives. The rest are ignored. These properties have the effect of defining classes of functionally equivalent stimulation, stimulation which for purposes of the theorist would be rendered indistinguishable. This stimulation may range from highly specific discrete behavior to rather continuous behavior (or stimulation) which takes place over a period of time. When this stimulation results from behavior, the term response (R) is used. Stimulus (S) refers to stimulation felt to effect behavior.
are formed in newsprint and on the television screen, the numerals 0-9 in arithmetic, and the letters of the alphabet in English. We use the terms *stimulus situation* and *behavior*, respectively, to distinguish such complexes from unanalyzed stimuli and responses.

**Definition 1:** An atomic stimulus is a symbol (indivisible set of properties) from some alphabet $A_s$.

**Definition 2:** An atomic response is a symbol from some alphabet $A_r$.

The alphabet may contain a finite or infinite number of elements. We let $A = A_s \cup A_r$.

If we restrict ourselves to visual stimulation, then, certain inputs (outputs) consist of linear sequences of atomic stimuli (responses). Other inputs (outputs) consist of two-dimensional arrays of atomic stimuli (responses)—and similarly for three-dimensional stimuli (responses). Normally, of course, not all arrays will be meaningful and of interest. Those inputs and outputs that are, are distinguished with the labels *stimulus situation* and *behavior*, respectively. (Some readers will note the analogy with formal languages.)

We can summarize this as

**Definition 3:** An input (output) array is a one, two, or three-dimensional finite arrangement (sequence) of atomic stimuli (responses).

An array $A$ is a subarray of array $B$ if and only if (iff) it is a proper part of array $B$.

**Definition 4:** A stimulus situation is an input array which is of interest to the theorist.

**Definition 5:** A behavior is an output array which is of interest to the theorist.

---

*Atomic* stimuli and responses may in principle be thought of as determined by taking the intersection of the partitions imposed on the real world by each criterion imposed by the theorist. Intuitively, stimuli (responses) may be thought of as atomic if and only if their components cannot be distinguished by any criterion of even potential interest to the theorist. The stimulus $24_{+56}$ is not likely to be atomic in this sense in arithmetic because the observer will almost certainly want to distinguish among the various digits of which the stimulus is composed.

*E.g.*, where all stimuli and responses in $\mathcal{J}: \mathcal{R}$ are atomic.

In this more technical use of the term, we distinguish between "behavior" (singular) and "behaviors" (plural).
The phrase "of interest to the theorist" is admittedly vague. In the next section we shall see that identifying a stimulus situation (behavior) as being of interest implies the availability to an observer of a perceptual (decoding) rule for detecting (constructing) it. Ordinarily, the scientific observer will concern himself primarily with those stimulus situations (behaviors) that can be detected (constructed) via rules commonly available in a given culture or subject population.

**Definition 6:** Two stimulus situations (behaviors) are *functionally distinct* iff they are *not* identical arrays.

Notice that just because one observable is functionally distinct from another does not necessarily mean that they have nothing in common. The stimulus situations "54" and "56," for example, have a "5" in common.

Because any given environmental event may give rise to more than one functionally distinct stimulus situation (behavior), stimulus situations (behaviors) may be thought of as specifying both that which is relevant and that which is irrelevant in the situation. The functionally distinct stimulus situation $^{24}_{+56}$, for example, specifies that "2," "4," "+," "5," "6," and "__," together with the structural relationships among them, are relevant; whereas other aspects of the environment—for example, the context in which the stimulus situation appears—are not relevant. This property of functionally distinct stimulus situations is important to keep in mind later on, where one or more functionally distinct stimulus situations (e.g., $^{2}_{+4}$ $^{4}_{6}$) are embedded in what for other purposes is itself a stimulus situation (e.g., in $^{24}_{+56}$). We summarize this point by:

**Definition 7:** A stimulus situation (behavior) $S'$ is a *substimulus* (subbehavior) of $S$ iff it is a subarray of $S$.

To complete our specification of observables, we need:

**Definition 8:** If $S$ is the stimulus situation for behavior $B$, then the $S$ and $B$ are said to form an *S-B pair* or *instance*.\(^7\)

For convenience, we sometimes also use the term "behaviors" to denote instances—for example, in the phrase "The set of rules introduced to account for a class of behaviors (i.e., S-B instances)."

The basic data of a theory of knowledge is a given set $\mathcal{S}:\mathcal{B}$ of S-B pairs. Finally, we observe

---

\(^7\)Note that $S$ is used to denote both stimulus and stimulus situation. Since we shall almost always refer to (S-R) or (S-B) instances, the intended meaning will always be clear from context.
**Definition 9:** If each stimulus situation in $S$ is atomic and each behavior in $B$ is atomic, then we denote the given set $S : H$ and refer to its elements as S-R instances.

**Examples.**

(1) Let $A = \{a, B, 0, 1\}$

$S = \text{the set of strings of the form } \text{"zB" where } z \text{ is a string of a's (e.g., aaaaaB)}.$

$B = \text{the set of strings of the form } \text{"Bw" where } w \text{ is a binary numeral (e.g., B101)}.$

In $S : B$ the output for any input $zB$ is $Bw$ where $w$ is the binary numeral representing the number of a's in $z$.

In this case, $B$ is the only atomic stimulus (response), but given any pair of distinct stimuli, one is a substimulus of the other (e.g., $aaB$ is a substimulus of $aaaaaB$).

(2) Let $A = \{0, 1, \ldots, 9, +, -, \times, \div\} \cup \text{Aux}$ where Aux is a finite set of auxiliary signs.

$S : B = \{S-B \mid S \text{ is a computational problem (e.g., 95 x 4) and } B \text{ is its solution (i.e., 380)}\}.$

In this case there are no meaningful atomic stimuli although 0, 1, ..., 9 may serve as atomic responses.

(3) Let $A = \{w \mid w \text{ is a word in English}\} \cup \{S\} \text{where } S \text{ is a nonterminal symbol }$.

$S : B = \{S-B \mid B \text{ is an English sentence}\}.$

2. RULES

Certain input-output pairs naturally belong together in the sense that the theorist "knows" at least one rule which applies to each input and generates each of the corresponding outputs. A set of such instances is referred to as the extension of a rule; and a description of the rule, as the intension (cf. Rogers, 1967; Scott, 1967). As we shall see below, any (denumerable) number of intensions may have the same extension. For present purposes, we distinguish further between the semantics of rules.

---

8 In recursive function theory (e.g., Rogers, 1967; Scott, 1967), intensions are formulated as descriptions. Reference to syntax is important in behavioral science for certain purposes (e.g., see the next section) but alone it does not appear to be sufficient. The descriptive approach provides no natural way to determine, for example, to what degree a subject knows and can use a particular rule.

The main purpose of this section is to define semantics in terms of extensions. In the process, a number of related ideas are also formalized. The basic idea parallels that of the previous section. We take certain rules as basic, equate their extensions and semantics, and construct all of the remaining rules in terms of them. This amounts to specification of some maximal level of detail beyond which further refinement of a rule is unwarranted. If the theorist's concern is with the mechanics of arithmetic, for example, the appropriate level of analysis may involve rules whose basic steps include rules for generating number facts. We shall see below that there is a close relationship between atomic stimuli (responses) and basic (what we shall call atomic) rules. In particular, once the atomic stimuli (responses) are identified, there is a minimal level beyond which a rule cannot be refined (i.e., broken down). Thus, for example, if the numerals 0, 1, 2, ..., 9 are taken as atomic, then the basic number facts correspond to minimal atomic rules. Any further "breakdown" of such atomic rules (e.g., to specify how the number facts are actually generated) would necessarily lead to consideration of more environmental detail—e.g., representation of the numerals as successors (of zero).

In any case, by starting at an atomic level, a strictly extensional approach to semantics is possible. We start by defining certain minimal (atomic) extensions, and then construct the semantics of more complicated rules from these minimal extensions. Our development shares certain characteristics with Engeler's (1968) treatment of algorithmic bases but is more general (although less formal).

Rather than begin with our more general version, we first develop the idea for \( L : A \) and then show how it can be generalized to \( L : B \).

**Definition 10:** An atomic operating rule (atomic or) is a set of input-output (I-O) pairs in which each input has a unique output.

\[
\text{atomic or} = \{ (I_i, 0_i) \mid 0_i = 0_j \text{ whenever } I_i = I_j \}.
\]

The set of inputs of an atomic operating rule is called its domain (Dom) and the set of outputs, its range (Ran).

9 The existence of some such minimal level is guaranteed by the fact that each rule is essentially equivalent to a Turing machine.

10 The theorist who chooses to think of a stimulus (e.g., numeral) as an indivisible whole is not likely to be concerned with how its response is derived from it. Conversely, if he were interested in the nature of the derivation, he would have to deal with particular properties of the stimulus, and effectively treat it as a composite.
Atomic operations correspond to classes of S-R instances in which the behavioral scientist (as distinct from the competence theorist) is not interested either in distinguishing among the instances or in the details of a (rule-based) account of these instances. Depending on the interests of the theorist, any of the following might be considered atomic:

1. knowledge of the addition facts through $9 + 9 = 18$,
2. the ability to carry in addition,
3. the ability to read (where reading ability is involved in the behavior of concern but is not itself the object of study and can safely be assumed of the subjects—e.g., college students),
4. the ability to compute.

**Definition II:** An atomic decision making capability (atomic dmc) is an ordered finite partition \( \{P_1, P_2, \ldots, P_n\} \) of a class \( C \) of elements \( E \) where \( P_i \) is a set of the form \( \{E \mid E \text{ satisfies predicate } P_i\} \) for \( i = 1, 2, \ldots, n-1 \) and \( P_n \) is the complement of \( \bigcup_{i=1}^{n-1} P_i \) relative to \( C \).

In an automaton, each atomic decision making capability (partition) contains exactly two equivalence classes (i.e., the branches in the state diagram of every automaton are binary with input elements 0 and 1).

Other examples of \( \text{dmc} \)'s are:

1. \( \{(x \mid x > 3), \{x \mid x \leq 3\}\) where \( C = \{x \mid x \text{ is a real number}\} \)
2. \( \{(n \mid n = 2n - 3), \{n \mid n > 2n - 3\}, \{n \mid n < 2n - 3\}\) where \( C = \{n \mid n \text{ is an integer}\} \)

Some observations:

1. Each element in the class \( C \) associated with a given atomic \( \text{dmc} \) is in exactly one element (i.e., equivalence class) of the atomic \( \text{dmc} \).
2. Atomic \( \text{dmc} \)'s correspond to decisions in computer programming and play an essential role in the theory. They make it possible to move from (operating) rules with finite domains to rules with infinite domains.

Physically speaking, this effects a change from stimuli which can be represented by bounded stimulus arrays to stimuli of arbitrary size. While not critical in the idealized (memory-free) theory proposed later on, stimulus size will almost certainly be a factor where memory and other physiologically based characteristics of the human are taken into account.

3. Notice the close relationship between an atomic \( \text{dmc} \) and the behavior expected of subjects in the typical concept attainment study where subjects are required to sort elements (i.e., stimuli) in a given universe into two piles, one containing exemplars of the concept (i.e., stimuli which satisfy a predicate) and the other containing nonexemplars (i.e.,
stimuli which do not satisfy the predicate. (Note: the sorting operation itself is due to a pair of constant valued rules which act on the stimuli in respective elements of the partition formed.)

4. Because of the close relationship between atomic $dmc$'s and predicates, we frequently suppress the difference and speak of the predicates (relations) themselves. This is particularly common in logic and computer programming where binary partitions (involving truth or falsity and yes or no, respectively) are common.

Rules are defined in terms of atomic $or$'s and $dmc$'s.

**Definition 12:** A rule $r$ is a labeled finite directed graph whose "arrows" are atomic $or$'s labeled with predicates and whose "nodes" are atomic $dmc$'s. One node is labeled "Start." Each node which contains predicates not used to label arrows emanating from it is labeled "Halt."

A stimulus $S$ and a rule $r$ together determine a response $R$ as follows: The $dmc$ labeled Start is applied first. The atomic $or$ applied next is the one labeled with that predicate in the starting $dmc$ which contains $S$. Control then flows to the $dmc$ determined by the preceding arrow (i.e., at the head of the arrow). In a similar manner, control proceeds from the nth atomic $dmc$ to the designated nth atomic $or$, to the $n+1$st atomic $dmc$. This continues until the computation terminates. This occurs where an atomic $or$ is undefined on a stimulus or no next atomic $or$ is specified.

**Definition 13:** An atomic rule (atomic $r$) is a rule consisting of one arrow, with one Start and one Halt node.

**Definition 14:** A subgraph $A$ is called a subrule of rule $B$ if the subgraph has a unique entrance node from the complement of $A$ in $B$, and becomes a rule upon proper labeling. Specifically, the entrance is labeled "Start," and each node which contains predicates not used to label arrows emanating from it is labeled "Halt."

**Definition 15:** Rule $A$ is a closed subrule of rule $B$ if it is a subrule of $B$ in which there is exactly one node designated "Halt."

We denote rules schematically in terms of modified flow diagrams. For example, see the diagram on the opposite page.

Each rule, then, corresponds to a computer program, where the arrows and nodes correspond to (labeled) operation instructions and test instructions, respectively. In our illustration, $dmc_1$ (Start) corresponds to "If $m = 0$, do $or_{1}$; else do $or_{2}$;" $or_{1}$ corresponds to "$(n,0) \rightarrow n$;" $or_{2}$
to "(n,m) → (n+1, m-1)," and \( \delta \) to Halt.

Furthermore, closed subrules correspond to subroutines in computer programming. Indeed, any rule \( A \) can be redefined in terms of closed subrules as long as each atomic rule in \( A \) is in exactly one closed subrule of \( A \). In this case, the "arrows" of the redefined rule correspond to closed subrules. A set of closed subrules with this property is called a complete disjoint set.

Definition 12 is incomplete in an important sense; it ignores both "perceptual" and "decoding" processes. It is simply assumed that stimuli are somehow automatically encoded (via "Start"), and responses automatically decoded (via "Halt"). Nothing is said about the details of either process. This is no great loss where a rule is designed to account for a given class of S-R instances. In this case, the stimuli and responses are unitary so that nothing essential can be gained by including perception and decoding in an account. If the stimulus is the display \( \begin{array}{ll} 2 & 4 \\ + & 3 \end{array} \begin{array}{ll} 6 & +3 \end{array} \) for example, then a rule which generates a response from this stimulus must do so by acting on the stimulus as a whole. The first step in an addition rule, for example, \( \frac{24}{6} \rightarrow \frac{36}{6} \) and the last \( \frac{24}{6} \rightarrow +3 \frac{6}{6} \) might be represented \( \frac{24}{6} \rightarrow +3 \frac{6}{6} \) and the last \( \frac{24}{6} \rightarrow +3 \frac{6}{6} \).

Most would agree, of course, that this is the stimulus and that this stimulus generates the indicated response. But in reality, it is typically not the whole stimulus which elicits the response associated with the stimulus (e.g., the two units digits elicit \( \frac{10}{10} \)). Furthermore, the response (behavior) is really just 60 and not the entire complex. Any complete account must reflect these details.

Fortunately, perception and decoding can be dealt with by extending our concept of rule. The essential idea involved is that of incorporating (atomic) capabilities into each rule which extract atomic stimuli (or, more generally, substimuli) from stimulus situations, and which construct
behaviors from atomic (sub)behaviors.

In the illustration above, for example, a verbal account might go as follows: (1) Locate and encode the two digits on the right—notice that the "response" here is a movement and (automatic) encoding of a substimulus (consisting of two atomic stimuli). (2) The encoded digits, then, serve as a (sub)stimulus for the (sub)response 10. (3) (Automatically) decode 10 by putting "0" beneath the units column and retaining the 1 (or placing it slightly above and to the left of the "0"). (4) This subresponse (is automatically perceived and) serves to elicit a move to the tens column. (5) Here, the tens digit from the first summation, together with the new tens digits, constitute the stimulus for the response 6. (6) Finally, "6" is placed in the tens place of the answer.

From this example it might appear that encoding (decoding) simply involves internalizing (externalizing) some substimulus (subbehavior). This is not entirely true, however, as can be seen by considering △ and △. Both displays are typically encoded as triangle. Triangle, however, is not a substimulus but rather a property which defines a class of stimulation in the real world. What we have called a substimulus (stimulus) is in actuality a canonical representative of a class of real world stimulation. Thus, the substimulus "5" in "45," for example, represents a class of equivalent displays (e.g., {5, 5, five, ...}). We shall see at the end of Section 5 that perceptual knowledge grows as rules interact, and that this growth has the effect of imposing finer and finer partitions on the environment.

Notice finally that when encoding and decoding rules are added to our formulation, stimuli and responses (behaviors) rarely serve as inputs and outputs, respectively, for the atomic or's in a rule. What we call stimuli and behaviors can, however, be recovered from arbitrary rules and that, in part, is what the formalization below is designed to make possible.

We strengthen our formulation by adding two kinds of atomic capability.

**Definition 16:** An atomic encoding (perceptual) rule (atomic p rule) is a set of input-output pairs with a finite Range in which each input is a stimulus and each output is a class of stimuli which includes the stimulus.

Since atomic p rules effectively partition the environment, they may alternatively be defined in that way (i.e., as partitions). Intuitively, atomic p rules involve the direct identification of stimulus properties. Where a property is a substimulus, the p rule may be thought of as locating the substimulus in the environment, and then, inserting the substimulus
in a class of stimuli. ( Appropriately restricted atomic p rules 
correspond to movements of reading heads of Turing machines together with 
their basic capability of "reading" 0's and 1's on tapes.)\(^{11}\)

Each of the following are atomic p rules: (1) a rule which partitions 
a class of objects according to color (size, shape, etc.), (2) a rule 
which partitions a class of numerals according to the number of digits, 
(3) a rule which "reads" the digits 0,1,2,...,9.

Definition 17: An atomic decoding rule (atomic d rule) is a finite 
set of input-output pairs in which each input is a class of (sub)- 
behaviors and each output is a (sub)behavior in this class.

Intuitively, atomic d rules involve altering the environment by con- 
structing observables so that the environment has specified properties. 
Where the property is a subbehavior, the d rule involves selecting the 
subbehavior (which is a canonical representative) and constructing it in 
the environment in a particular location. (Atomic d rules correspond in 
Automata theory to movements followed by "writing" on tapes.)

Each of the following are atomic d rules: (1) a rule which constructs 
objects having specified shapes (colors, sizes, etc.), (2) a rule which 
constructs a numeral with a specified number of digits, (3) a rule which 
"writes" the digits 0,1,2,...,9 in particular locations.

By way of summary, then, rules as previously defined operate exclusive- 
ly on encoded stimuli and generate undecoded responses. Adding atomic p 
rules and atomic d rules to our formulation of rules has the effect of 
adding "arrows" to our directed graph characterization, which involve in- 
sertion into classes (encoding) and extraction from classes (decoding), 
respectively. Thus, what was an atomic rule before involves three opera- 
tions—an atomic encoding (perceptual) rule, an atomic operating rule, and 
an atomic decoding rule. (Strictly speaking, atomic operating rules must 
be restricted to finite domains unless looping of atomic p rules is allowed, 
because the ranges of the latter are finite.) In the study of perception, 
encoding rules may be expected to play the primary role. Decoding rules 
would appear critical in studies involving skills of various kinds (e.g., 
speaking, writing).

Encoding rules make it possible to spell out the manner and order in 
which substimuli are "extracted" from the environment. It is no longer 
necessary to assume that all stimuli are extracted automatically as wholes.

---

\(^{11}\) As we shall see in discussing the idealized (memory-free) theory, 
operational tests of encoding rules may be obtained by requiring subjects 
to classify stimuli. In decoding, the subject is required to construct 
responses of various types.
In a like manner, decoding rules make it possible to detail the process by which responses are generated. Responses typically are not generated instantaneously, but are constructed from subresponses over a period of time. (As an exercise, devise a precise directed graph characterization of the verbal account of addition. That is, formulate a general rule for adding numbers which includes both p and d rules.)

In order to place the role of (atomic) p and d rules in perspective, it is instructive to consider the rule \((\frac{A + L}{2})N\) for summing arithmetic series. (A is the first term of the series, L the last, and N the number of terms.) Observe that identifying A and L is essentially a matter of locating them. (This is not entirely true, unless A and L are bounded from above.) A complete account of how N is determined, however, requires more than just encoding (insertion into classes). A and L are substimuli, whereas N is a property derivable from substimuli. (One possible rule might involve encoding the terms one by one and counting.) It should be emphasized, however, that in many applications this distinction is unimportant and may be suppressed.

Although the atomic p (and d) rules in a rule generally speaking do not operate directly on stimulus situations (nor directly generate behaviors), it is always possible to recover the stimuli (and behaviors) associated with a rule given only the substimuli—inputs (and subresponses—outputs) associated with each of its constituent atomic p (and d) rules. That is, stimulus situations and behaviors are implicit in any rule which accounts for them. Given a particular rule, a stimulus situation associated with the rule may be determined by tracing through the rule as follows: Go to the starting dmc. Pick one element in one equivalence class that belongs to the domain of some atomic or. Construct (in the environment) a canonical representative of this element. Each time an encoding rule is encountered, move to the appropriate location in the environment and construct a canonical element in the output class (of the encoding rule). (Notice that elements in the extensions of p rules, or's, and dmc's are classes of observables.) The process is repeated until the computation terminates. The configuration of substimuli which results is defined to be a stimulus situation associated with the rule. The domain of a rule r is the set of stimulus situations associated with the rule.

Behaviors are generated by simply applying the rule to stimulus situations. That is, the behavior associated with a rule r and stimulus situation S is determined by applying r to S. The behavior is the configuration formed from the subresponses of the atomic d rules in r. The range of a rule, then, is the set of behaviors generated by application of the rule.
to its domain.

In effect, it is possible not only to build up rules out of atomic rules but also to detail how stimuli (behaviors) may be recovered from rules. Recall that in Section 1 we promised to specify more precisely the sense in which certain arrays might be of interest. Such interest is a direct function of the availability of appropriate perceptual (and decoding) rules in a given culture or subject population.

We summarize this in

**Definition 18:** The extension \( \text{Ext}(r) \) of a rule \( r \) is the set consisting of all S-B pairs such that S is in the domain of the rule and B is the output generated by application of \( r \) to \( S \).

Notice that any number of rules may have the same extension. For example, consider the borrowing and equal additions methods for subtraction. The extension of an atomic rule, however, is essentially (up to one-to-one correspondence) the same as the set of input-output pairs of the constituent atomic \( \text{or} \). (The extension of the atomic rule by definition consists of S-B instances, whereas the set of input-output pairs of the atomic \( \text{or} \) includes the encoded counterparts of these S-B pairs.)

A rule is said to be *finitary* if and only if its extension contains a finite number of S-B instances. Rules with infinite extensions may be built up from (finite numbers of) finitary rules by looping back on the constituent rules in the usual way.

Our final point is that rules can often be refined by considering processes underlying one or more of their atomic rules. The associated \( \mathcal{S} : \mathcal{B} \) classes of S-B instances, however, place limits on the degree of refinement possible. The substimuli involved in any rule account involving perception, for example, may not involve discriminations finer than the atomic stimuli. In order to add more detail it would be necessary to redefine the given set of S-B instances.

More generally, we state

**Definition 19:** (a) Rule \( r' \) is a refinement of rule \( r \) if there is a one-to-one correspondence between the atomic operation (p, \( \text{or} \), and d) rules of \( r \) and the closed subrules of \( r' \) such that the extension of each atomic rule in \( r \) is the extension of the corresponding closed subrule of \( r' \). (b) Rule set \( K_1 \) is a refinement of rule set \( K_2 \) if there is a one-to-one correspondence between the rules in \( K_1 \) and \( K_2 \) such that each rule in \( K_1 \) is a refinement of the corresponding rule in \( K_2 \).
Observation: If \( r' \) is a refinement of \( r \), then the extension of \( r' \) equals the extension of \( r \) (i.e., \( \text{Ext} (r') = \text{Ext} (r) \)).

To see this we observe that the atomic \( p \) rules in \( r \) correspond in one-to-one fashion with closed \( p \) subrules of \( r' \) with the same extensions. But, the domains of rules are determined uniquely by the extensions of their \( p \) rules. Also, rules \( r \) and \( r' \) generate exactly the same outputs for any given input since the closed subrules of \( r' \) have exactly the same effect as the atomic operation rules of \( r \). Hence, \( \text{Ext} (r') = \text{Ext} (r) \).

In thinking about perception (decoding), it is important to recognize that different perceptual (decoding) rules may operate at different levels. Specifically, in Section 5 we shall see that perceptual rules generated (by application of higher order rules) may involve a finer basis (i.e., sub-substimuli) than the perceptual rules from which they are generated. Later on, it will become apparent that learning to perceive in general involves moving from gross to finer distinctions concerning the environment. According to Piaget, for example, the young child is able to distinguish between figures which are and are not (topologically) closed before he is able to distinguish say, between, squares and circles, both of which are closed.

3. PROGRAMS: DESCRIPTIONS OF RULES

Although rules may serve as inputs and outputs of other rules (cf. Section 4), rules themselves may not serve as stimulus situations or behaviors. Observable counterparts of rules do exist, however, and these we shall call programs. Programs essentially are descriptions of rules in a suitable language.

For present purposes we shall simply assume that some suitable language exists, say ordinary mathematical English or some programming language. Modifying Scott (1967) slightly, this may be accomplished by introducing \( n \)-tuples of predicate terms \( \langle P_1, \ldots, P_n \rangle \) for the atomic decision-making capabilities, and function terms \( \langle F, E, D \rangle \) for atomic operating, encoding, and decoding rules, respectively. In addition, we need labels \( (L) \), and the following special terms: Start, Go to, If, Then, Do, Else, ;, ;, and Halt.

**Definition 20:** An instruction is a string of one of the following six forms

\[
\begin{align*}
\text{Start}: & \quad \text{Go to } L \\
L: & \quad \text{Do } E; \text{ go to } L' \\
\end{align*}
\]

(start instruction)

(encoding instruction)
L: Do F; go to L' (operation instruction)
L: If \( P_1 \) then go to \( L'_1 \);
    if \( P_2 \) then go to \( L'_2 \), ...;
    if \( P_{n-1} \) then go to \( L'_{n-1} \);
    else go to \( L'_n \). (test instruction)
L: Do D; go to L' (decoding instruction)
L: Halt (halt instruction)

where \( L, L', \text{ and } L'_i \) are members of the set of labels \( \mathcal{L} \); \( F, D, \text{ and } E \) are function terms in \( \mathcal{F}, \mathcal{K}, \text{ and } \mathcal{C} \), respectively, and \( P_i \) is a predicate term in \( \mathcal{P} \).

**Definition 21:** A program is a finite set of instructions containing exactly one start instruction and containing for each label that occurs anywhere in any instruction in the program, exactly one instruction that begins with that label.

In our system it will turn out that programs may denote rules which act on classes of programs (as inputs). The instruction symbols of which such programs are composed, therefore, will be of an essentially higher order; the symbols denote operations and predicates which act on classes of predicate and function symbols. The total number of predicate and function symbols needed in any particular (finite) account, of course, will always be finite.

The distinction between rules and programs can be suppressed in many applications where the theorist is uninterested in the rules used in interpretation (e.g., in encoding symbols, and then assigning them meanings) and description (e.g., in representing meanings, and then decoding the representations). In simple arithmetic computation, for example, no distinction is usually made between numbers and numerals (number names). In this case it is particularly convenient to assume for each higher order rule that it includes just one atomic rule which combines encoding and interpretation, and one atomic rule which combines description and decoding.  

4. KINDS OF RULES

**Definition 22:** Simple rules are rules whose extensions do not include stimuli or responses that are programs for other rules.

---

12Where a distinction between rules and programs is desired, there are two kinds of relevant data (observables); one involves paraphrasing and the other, application of rules corresponding to programs (cf. Chapter 7, Section 5).
**Definition 23:** Higher order rules (ho rules) are rules that are not simple; their extensions contain stimuli and/or responses that are programs for other rules.

Although it is possible to conceive of ho rules which operate exclusively at the syntactic level (i.e., which make no reference to meaning), such rules have played a central role in linguistic grammars (cf. Chomsky, 1965) and are not of central concern here. For present purposes, it is more natural to think of ho rules as actually interpreting the programs on which they operate. We shall not, however, have much to say in this section about the details of this process (See Scandura, 1970b, and Chapter 7 for further discussion). It is assumed that encoding and interpretation (and description and decoding) are combined as indicated above. In this case, it is possible to discard the initial and terminal atomic rules with no loss of generality and just talk about rules which operate on and generate rules (i.e., meanings). This practice is adopted throughout this section, except where we discuss ho rules involved in perception. In particular, when we refer to the extension of a ho rule we mean the set of input rule-output rule pairs.

The main purpose of this section is to identify some of the basic kinds of ho rules. No claim is made regarding the exhaustiveness of these basic types, however.\(^{13}\)

**Definition 24:** A composition rule c is a rule which maps pairs of rules represented by

\[
\begin{align*}
\text{Start} & \quad \langle P_1, \overline{P}_1 \rangle \quad r_1 \quad \langle P_r \rangle \quad \text{Halt} \\
\text{and} & \\
\text{Start} & \quad \langle P_2, \overline{P}_2 \rangle \quad r_2 \quad \text{Halt}
\end{align*}
\]

into composite rules represented by

\[
\begin{align*}
\text{Start} & \quad \langle P_1, \overline{P}_1 \rangle \quad r_1 \quad \langle P_2, \overline{P}_2 \rangle \quad r_2 \quad \text{Halt}
\end{align*}
\]

\(^{13}\)Nonetheless, these types do form a basis for a wide variety of other types. Further, it appears (although we shall not attempt to prove it) that a relatively small subset of the identified types may be sufficient for generating the others in much the same way that "not" and "and" form a basis for generating all of the basic connectives used in the statement logic (e.g., "or," "implies").
For simplicity, we define $c$ as a rule with an extension of the form
\[ c = \{(r_1, r_2; r_3) \mid \text{Ran } r_1 \subseteq \text{Dom } r_2 \text{ for all } (r_1, r_2) \in \text{Dom } c \text{ and Dom } r_3 = \text{Dom } r_1 \text{ and Ran } r_3 = \text{Ran } r_2 \}\]

Composition rules act on pairs of rules in which the outputs of one serve as inputs for the other. Because the individual rules $r_1$ and $r_2$ shared common labels some relabeling was necessary in forming the composite. (This can always be accomplished in an algorithmic fashion.) In view of what follows, it may also be worth noting that the $\partial \mathcal{C}$'s of the nodes in the composite rule are identical to (certain of) the $\partial \mathcal{C}$'s of the nodes in $r_1$ and $r_2$. This is not always the case (cf. Definition 26).

**Definition 25:** The inverse $c'$ of a composition rule $c$ undoes the composition rule. That is, the extension of $c'$ is of the form
\[ c' = \{(r_1 \circ r_2, r_2; r_1)\}\]

**Definition 26:** A simple generalization rule $sg$ is a rule whose extension is of the form
\[ sg = \{(r_1, r_2; r_3) \mid r_1(S) = r_2(S) \text{ for all } S \in \text{Dom } r_1 \cap \text{Dom } r_2 \text{ and } (r_1, r_2) \in \text{Dom } sg; \text{ also Dom } r_3 = \text{Dom } r_1 \cup \text{Dom } r_2 \text{ and } r_3(S) = r_1(S) \text{ if } S \in \text{Dom } r_1; \text{ } r_2(S) \text{ otherwise}\}\]

Simple generalization rules act on pairs of rules and generate equivalent rules with domains that are unions of the given ones. It is important to notice in this connection that the $\partial \mathcal{C}$ in the initial node of $r_3$ is different from any of the $\partial \mathcal{C}$'s associated with $r_1$ or $r_2$. In particular, this (initial) $\partial \mathcal{C}$ is formed by forming a union and accepts any stimulus that is accepted by either the initial $\partial \mathcal{C}$ of $r_1$ or the initial $\partial \mathcal{C}$ of $r_2$. That is, the initial $\partial \mathcal{C}$ is $\{\text{Dom } r_1 \cup \text{Dom } r_2, \text{ Dom } r_1 \cup \text{Dom } r_2\}$.

**Definition 27:** A conjunction rule $cj$ is a rule whose extension is of the form
\[ cj = \{(r_1, r_2; r_3) \mid \text{Dom } r_1 = \text{Dom } r_2 \text{ for all } (r_1, r_2) \in \text{Dom } cj; \text{ also Dom } r_3 = \text{Dom } r_1 = \text{Dom } r_2 \text{ and } r_3(S) = r_1 \times r_2(S)\}\]

For example, let $r_1$ add pairs of whole numbers and $r_2$ multiply pairs of whole numbers, then $r_3$ adds and multiplies pairs of whole numbers (i.e., the outputs are pairs consisting of sums and products). It would be a simple matter to generalize $cj$ by allowing the initial $\partial \mathcal{C}$ of $r_3$ to differ from the $\partial \mathcal{C}$'s of $r_1$ and $r_2$. In particular, by letting this initial $\partial \mathcal{C}$ be $\{\text{Dom } r_1 \cap \text{Dom } r_2, \text{ Dom } r_1 \cap \text{Dom } r_2\}$, $cj$ can be defined on arbitrary pairs of rules with nonempty intersections. (Intersection $\partial \mathcal{C}$'s play a central role in learning how to perceive, and as we shall see in Section 5,
they are crucial in accounting for $\mathcal{F}: \mathcal{B}$ classes.)

**Definition 28:** An elimination rule $el$ is a rule which eliminates unnecessary subrules of a rule. More specifically, the extension of an elimination rule is of the form

$$el = \{(r, r-r') | r \text{ contains closed subrule } r' \text{ whose inputs and outputs are identical and } r-r' \text{ is the rule obtained by "detaching" } r' \text{ from } r\}$$

For example, represent $r = \frac{2n+1-1}{3}$ by

$$\begin{array}{c}
\xymatrix{
\text{n} \ar[r]^{x^2} & 2n \ar[r]^{+1} & 2n+1 \ar[r]^{-1} & (2n+1)-1 \ar[r]^{\text{assoc}} & 2n+(1-1) \ar[r]^{\text{identity}} & 2n \ar[r]^{/3} & 2n/3
}
\end{array}$$

where the circles are labeled nodes.

Then, for some suitable $el$, $el(r)$ is represented by

$$\begin{array}{c}
\xymatrix{
\text{n} \ar[r]^{x^2} & 2n \ar[r]^{/3} & 2n/3
}
\end{array}$$

In the following, $r$ is a rule and $p$, a subrule in $r$ for determining stimulus properties (i.e., for perception). Hence, rule $r$ may be denoted $r = p + (r - p)$.

**Definition 29:** A restriction rule $res$ is a rule whose extension is of the form

$$res = \{ (p + (r-p), p'; p' + (r-p)) | p \text{ and } p' \text{ are perceptual rules such that } \text{Dom } p' \subseteq \text{Dom } p \}$$

For example, let $r = p_N + (r - p_N)$ be a rule for summing arbitrary arithmetic number series (e.g., $N \{(A + L)/2\}$ where $p_N$ is a perceptual rule for finding the number of terms $N$ for arbitrary arithmetic series (e.g., $p_N = (D + L - A)/D$ where $A$ is the first term of the series, $L$ is the last, and $D$ is the common difference between terms). Also, let $p'_N$ be a perceptual rule for finding $N$ for arithmetic series of the form $1 + 3 + 5 + ... + (2N-1)$ (e.g., $p'_N = \frac{A+L}{2}$). (Presumably, $p'_N$ is more "efficient" than $p_N$ on its restricted domain.) In this example, $r$ may be represented by $[(D + L - A)/D] [(A + L)/2]$ and $p'_N + (r - p_N)$ by $\left[\frac{A + L}{2}\right] [(A + L)/2]$. For an experimental study involving similar rules see Scandura and Durnin (1968). (This experiment is summarized in Volume II, Chapter 2.)

14Restriction rules appear to be equivalent to the inverse of a composition rule followed by composition. A similar comment applies to generalization rules (Definition 30).

15Elimination and restriction rules may play important roles in increasing efficiency of behavior. In particular, practice on a (continued)
**Definition 30:** A generalization rule $g$ is a rule whose extension is of the form

$$g = \{(p' + (r-p), p; p + (r-p)) \mid p \text{ and } p' \text{ are as above}\}$$

For example, again consider the general rule $N [(A+L)/2]$ for summing arithmetic series and its restriction $N' [(A + L)/2]$ (where $N'$ is determined by $\frac{A+L}{2}$ and $N$ by $[(D + L - A)/D]$).

**Definition 31:** A selection rule $s$ is a rule whose extension is of the form

$$s = \{(r_1, r_2; r_1) \mid \text{Dom } r_1 = \text{Dom } r_2 \text{ and } r_1 = r_1 \text{ or } r_2\}$$

For example, one selection rule might be described "If $r_1$ was used on the previous trial, then select $r_2$; else $r_1". The selection rule used in a study reported in Chapter 8 was, "If the stimuli are edible objects, then select $r_1; else r_2'".

5. NATURE OF THE THEORY OF KNOWLEDGE

With this background, a theory of knowledge $\mathcal{K}$ may be defined as an $n + 3$ tuple

$$\mathcal{K} = \langle A, \mathcal{J} : \mathcal{G}, K, r_1, r_2, \ldots, r_n \rangle$$

where $A$ is a finite alphabet, $\mathcal{J} : \mathcal{G}$ is a class of pairs consisting of stimulus situations and their corresponding behaviors, and $K$ is a finite set of rules (which may include percepts (i.e., enclosed stimuli). The $r_i$, $i = 1, 2, \ldots, n$ are the rules in $K$. These rules may act on rules in $K$ as well as on the S-B in $\mathcal{J} : \mathcal{G}$. Notice also that programs (descriptions of rules) are potentially observable and may serve as elements of $\mathcal{J} : \mathcal{G}$ whereas rules themselves may not. Conversely, encoded percepts may belong to $K$ but stimuli may not.

15(cont'd.) given rule, especially under timed conditions, may lead to the elimination of unnecessary subrules and, hence, result in the acquisition of a more efficient rule. Presumably, the most efficient rules consist of an atomic encoding rule followed by an atomic decoding rule. Efficiency can sometimes be further increased by restriction because restricted rules (see above) frequently involve fewer operations.

Although only treated herein in passing, the generation of new perceptual rules (cf. Section 5) apparently serves a similar function, particularly with young children and/or novel stimulation.

Where both efficient and relatively inefficient rules are available for solving a given (kind of) task, selection rules (cf. Def. 31) presumably serve to insure that the more efficient rules are used where needed (e.g., under timed conditions).

16This rule corresponds to the "alternating" selection rule studied in Chapter 8.
Nonetheless, for many purposes involving complex knowledge, it is convenient to ignore the details of encoding and decoding processes. In this case, we have

\[ \mathcal{H} = \langle \mathcal{S}, \mathcal{A}, K, r_1, r_2, \ldots, r_n \rangle \]

Most of this section is devoted to making precise and illustrating what it means for \( K \) to provide an account of \( \mathcal{S}, \mathcal{A} \). The general idea (cf. Scandura, 1971a) is that the rules in \( K \) may operate on other rules in \( K \) to produce new rules. A rule set is said to account for a given S-R pair in \( \mathcal{S}, \mathcal{A} \), if a rule is eventually generated in this manner that yields \( R \) when applied to \( S \).

With this in mind, we introduce

**Definition 32**: \( K^1 = K \), and \( K^n = K^{n-1}(K) \cup K^{n-1} = \bigcup_{r \in K^{n-1}} \{ r(K) \cup K^{n-1} \} \)

where \( K^{n-1}(K) \) means the rule set generated by applying all of the rules in \( K^{n-1} \) to every element in \( K \) in its domain. The potential knowledge \( PK \) associated with a rule set \( K \) is

\[ PK = K^\infty = \bigcup_{n=1}^{\infty} (K^n(K) \cup K^n) \]

For example, consider the theory \( \mathcal{H}^{17} \) with alphabet \( \{ a, B, 0, 1 \} \), \( \mathcal{S} = \{ (xB, By) | x \text{ is a string of } a's, y \text{ is the binary numeral which represents the number of } a's \} \), and \( K = \{ r_1, r_2, \circ, i \} \) where \( r_1 = xxBy \rightarrow xB0y \), \( r_2 = xxaBy \rightarrow xBly \), \( \circ = r, r' \Rightarrow r \circ r' \) for all \( r, r' \) and \( i = r \Rightarrow r \) for all \( r \). (Note: \( \circ \) is called the generalized composition rule and \( i \) the identity. The n-fold application of \( \circ \) is denoted \( \circ^n \).) Then

\[
\begin{align*}
K^1 &= \{ r_1, r_2, \circ, 1 \} \\
K^2 &= \{ r_1, r_2, \circ, i, r_1 \circ r_2, r_2 \circ r_1, \circ^2 \} = K^1 \cup \{ r_1 \circ r_2, r_2 \circ r_1, \circ^2 \} \\
K^3 &= K^2 \cup \{ r_1 \circ r_2 \circ r_1, r_1 \circ r_2 \circ r_2, r_2 \circ r_1 \circ r_1, \ldots; \circ^3 \} \\
&\vdots \\
K^n &= K^{n-1} \cup \{ r_1 \circ r_2 \circ \ldots \circ r_1 \circ r_2 | i_j = 1 \text{ or } 2 \text{ for } j = 1, \ldots, n; \circ^n \}
\end{align*}
\]

In general, if \( K \) contains the identity rule \( i \), then \( K^n = K(K) \) for \( n \geq 2 \) and \( K^\infty = \bigcup_{n=1} K^n(K) \). We make this simplifying assumption in what follows.

Two comments are worth noting. First, \( K^n \) may be thought of as an upper bound on the knowledge that might possibly by acquired via the n-

---

17 I am indebted to Paul Rosenbloom for suggesting this example.
fold application of rules in $K$. PK is the asymptotic state of knowledge—that is, the knowledge which might be acquired by a knower characterized by $K$ given an indefinite amount of time (for rules to operate on and generate new rules in all possible ways). Second, in the idealized learning theory introduced in Chapters 7 thru 9, the growth of knowledge takes place in a more restricted, although closely related manner. A specified amount of knowledge (i.e., some number of rules—possibly zero) is added each time the knowing subject addresses a new task.

**Definition 33:** (a) A rule $r \in K^n$ is said to **account** for an S-R pair in $S : R$ iff $r(S) = R$.

(b) A rule $r' \in K^n$ is said to **account** for a rule $r'' \in K^{n+1} - K^n$ iff there are rules $r_1, r_2, \ldots, r_i$ in $K^n$ such that $r'(r_1, r_2, \ldots, r_i) = r''$.

In the above example, (a) $r_1 \circ r_1 \circ r_2 \in K^3$ accounts for $aaaaB \rightarrow B100$ because $aaaaB \rightarrow aab0 \rightarrow ab00 \rightarrow B100$. (b) $o^3 \in K^3$ accounts for $r_2 \circ r_2 \circ r_2 \circ r_1 \in K^4$ because $o^3 (r_2, r_2, r_1, r_1) = r_2 \circ r_2 \circ r_1 \circ r_1$.

**Definition 34:** (a) An S-R pair in $S : R$ is said to be **nth order generable** from a rule set $K$ iff there exists a rule $r \in K^n$ which accounts for the S-R pair.

(b) A rule $r''$ is said to be **nth order generable** ($n \geq 2$) from a rule set $K$ iff there are rules $r', r_1, r_2, \ldots, r_i$ in $K^{n-1}$ such that $r'(r_1, r_2, \ldots, r_i) = r''$.

(c) A rule is **first order generable** from a rule set $K$ iff it is in $K$. (Note: Rules may operate on given S-R pairs in determining $K^n$).

For example, (a) $aaaaB \rightarrow B100$ is third order generable because there is a rule $r_1 \circ r_1 \circ r_2 \in K^3$ which accounts for the pair. (b) Similarly, $r_2 \circ r_2 \circ r_1 \circ r_1$ is fourth order generable because $o^3, r_1$, and $r_2 \in K^3$ and $o^3 (r_2, r_2, r_1, r_1) = r_2 \circ r_2 \circ r_1 \circ r_1$.

**Definition 35:** A rule set $K$ in a theory of knowledge $H$ is said to account for $S : R$ iff for each S-R pair in $S : R$, there is a finite number $n$ such that S-R is $n$th order generable from $K$.

Evaluating a theory of knowledge empirically involves determining whether or not arbitrary S-R pairs in $S : R$ are derivable from $K$. Certain S-R pairs, of course, will be trivially easy to account for so that to make a convincing case for the theory one usually concentrates on aberrant cases or cases which otherwise make it possible to distinguish between
alternative theories. (This is standard practice in linguistics.)

Although $n$ is generally restricted to a finite number, presumably Definition 34 could be extended to allow for generability at asymptote (i.e., generability via rules derivable only at asymptote). It is not clear, however, where this might be useful, if at all. Another and possibly more useful alternative might be to restrict the size of $n$ (e.g., $n \leq 7$). This alternative could be of some value in applications where memory is a factor. For certain purposes it might also be useful to define the notion of a "uniform account" of $\mathcal{S}: \mathcal{R}$ in terms of a maximum $n$. In this case, we might look for conditions on $\mathcal{S}: \mathcal{R}$ and $K$ such that providing an account implies providing a uniform account. (For example, any account in a behavioral objectives type theory (see Section 6) is a uniform account.)

Some further definitions are

**Definition 36:** A theory of knowledge $\mathcal{K}$ is **simple** iff $K$ contains the identity.

**Definition 37:** A theory of knowledge $\mathcal{K}$ is **atomic** iff each rule in $K$ is atomic.

**Definition 38:** A theory of knowledge $\mathcal{K}$ is **finitary** iff each rule in $K$ is finitary (i.e., has a finite domain).

The example given above is simple and atomic, but not finitary.

**Definition 39:** Let $K$ and $K'$ be rule sets in Theories $\mathcal{K} = (\mathcal{S}: \mathcal{B}, K, r_1, \ldots, r_n)$ and $\mathcal{K}' = (\mathcal{S}: \mathcal{B}, K', r'_1, \ldots, r'_m)$ respectively. Then, $K$ and $K'$ are said to be **equivalent in computing power** iff every $S$-$R$ pair in $\mathcal{S}: \mathcal{R}$ either can be accounted for by both $K$ and $K'$ or by neither one.

For example, define $\mathcal{K}$ as above and let $K' = \{r'_1\}$ where $r'_1 = \text{Start} \xrightarrow{\text{r1}} \text{Halt}$. (The node can be described as "If $x \neq \emptyset$ and the string is of the form xxBy, do r$_1$; else do r$_2$. If $x = \emptyset$, Halt.")

It is instructive to consider a second example in which $K$ and $K'$ are identical except for the generalized composition rule. Although the extensions are the same, $K$ contains one composition rule ($\circ$) and $K'$ another ($\circ'$) where

\[ \circ = \text{Start} \xrightarrow{\text{dmc}} \text{or}_2 \xrightarrow{\text{r2}} \text{Halt} \]

\[ \text{or}_1 \rightarrow \text{Start} \]
Foundations of the Theory of Knowledge 109

and

\[ e' = \text{Start} \xrightarrow{o_1} \text{Start} \xrightarrow{o_2} \text{Halt} \]

with

\[ o_{r_1} = \text{Select two rules } r_1 \text{ and } r_2 \text{ (in } K \text{ or } K') \]

\[ o_{r_2} = \text{Form the composition } r_1 \circ r_2 \]

\[ \text{dmc} = (A = \{(r_1, r_2) | \text{Ran } r_1 \cap \text{Dom } r_2 \neq \emptyset, \overline{A}\}) \]

\[ \text{dmc}' = (A = \{(r_1, r_2) | \text{Ran } r_1 \cap \text{Dom } r_2 \neq \emptyset, S \in \text{Dom } r_1, \text{ and } R \in \text{Ran } r_2\}, \overline{A}) \]

The important thing to notice here is that \( \text{dmc}' \) (as opposed to \( \text{dmc} \)) makes direct reference to the S-R pair to be accounted for—i.e., S and R are among the arguments of \( o' \). Definitions 22-23 anticipated this possibility. Allowing S's and R's to enter in this way makes for more efficient search. Obviously inappropriate rules can be eliminated from consideration relatively quickly via decisions, rather than allowing a derivation to proceed only to find that the derived rule does not account for the given S-R pair. In short, such decisions may sharply reduce the number of false starts. With small rule sets such as \( K \) and \( K' \), of course, efficiency is not critical but it becomes increasingly so as rule sets become larger. Furthermore, as we shall see in discussing the idealized (memory-free) theory of learning in Chapters 7 thru 9 decisions involving S-R pairs (or equivalently, goal situations) seem to more adequately reflect what human subjects are likely to know and do.

Finally, we consider briefly what is involved in the more general form of the theory of knowledge which involves S-B (rather than S-R) pairs. In this case, we need to consider perception and decoding as well as the simple generation of responses from stimuli. Undoubtedly the easiest way to accomplish this would be to simply add on an atomic encoding and an atomic decoding rule to each rule in \( K \), where \( K \) is associated with a corresponding \( \mathcal{A} : \mathcal{B} \) class. More generally, of course, we must allow encoding and decoding rules to act on arbitrary substimuli.

The key question in this case is whether all of the respective encoding and decoding of stimuli and behaviors in \( \mathcal{A} : \mathcal{B} \) must be done directly (i.e., by one of the encoding or decoding rules attached to the rules in \( K \))—or, whether new encoding and decoding rules may be derived indirectly. In particular, are new perceptual and decoding rules derivable via higher order rules as in the simplified form of the theory above? And,

---

\(^{18}\)Strictly speaking, \( o_{r_1} \) is a nondeterministic operating rule.
if so, what are the specifics of the process? Although tentative because of the almost complete absence of relevant data, I believe that the above mechanisms account also for the growth of perceptual and decoding skills.

To see how this can be accomplished, we first note that atomic encoding and decoding rules involve stimuli and behaviors, respectively, and insert into or extract from classes (cf. Scandura, 1970b). Such rules do not distinguish among stimuli (behaviors) in any given class. For example, a rule may encode each of a class of stimuli as triangles without distinguishing between large and small ones.

Insofar as generating new encoding (decoding) rules is concerned, the important point is that different encoding (decoding) rules will generally divide up the "environment" (i.e., $\mathcal{S}$) in different ways. Hence, combining such rules via application of higher order rules may generate new encoding (decoding) rules. The basic process involves forming intersections of given $\mathcal{A}$'s (cf. the discussion following Definition 27). In effect, the rules in $K$ at any particular stage determine which stimuli may be distinguished. As knowledge grows (in $K^2$), finer and finer distinctions may be made. In devising a characterizing rule set, then, the theorist must not only make judgements concerning (internal) operations and decisions, but also make judgements about how the environment is to be initially partitioned.

The exact nature of the process is best seen in terms of an example. We consider a number of stimuli, each of which may be classified according to two independent dimensions.

<table>
<thead>
<tr>
<th>Letter First</th>
<th>Letter Second</th>
<th>Triangle</th>
<th>Circle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>a</td>
<td>#</td>
<td>Small</td>
</tr>
<tr>
<td>Large</td>
<td>A</td>
<td>#</td>
<td>Large</td>
</tr>
</tbody>
</table>

The behaviors are simply descriptions of the stimuli. For example, the behavior associated with "$\Delta$" is "Small Triangle." Thus

$$\mathcal{S} : \mathcal{B} = \{(a\#, \text{Small-Letter First}), \ldots , (\#A, \text{Large-Letter Second}), (\Delta, \text{Small Triangle}), \ldots , (\bigcirc, \text{Large Circle})\}$$

In this case, of course, it would be a simple matter to devise a rule set $K$ which accounts for $\mathcal{S} : \mathcal{B}$ directly. A more interesting possibility

---

19 Some of Piaget's work, however, is suggestive. His finding that young children are frequently unable to deal with two or more dimensions simultaneously seems to be particularly relevant.
is provided by

\[ K' = \{ \Theta, \{a\#, \#a\} \rightarrow \text{Small}, \{A\#, a\#\} \rightarrow \text{Letter First}, ... \] 

\[ \{\triangle, \bigcirc\} \rightarrow \text{Large}, \{\bigcirc, \bigcirc\} \rightarrow \text{Circle} \]

where \( \{x,y\} \) is the class containing stimulus \( x \) and stimulus \( y \), and \( \Theta \) maps each pair of rules (e.g., \( \{\triangle, \bigcirc\} \rightarrow \text{Large}, \{\bigcirc, \bigcirc\} \rightarrow \text{Circle} \)) into a rule whose domain is the intersection set and whose response is a conjunction (e.g., \( \{\bigcirc\} \rightarrow \text{Large Circle} \), or just \( \bigcirc \rightarrow \text{Large Circle} \)). (It is worth noting that \( p \) rules can partition the environment only to the level of atomic stimuli.)

It is easy to see that \( K' \) provides an adequate account of \( J : S \).

But, it is even more important to notice how naturally \( K' \) can be generalized to deal with larger \( J : S \) classes by introducing larger input sets (e.g., \( \{\triangle, \bigcirc, \square, ...\} \)) and/or new lower order rules (e.g., \( \{...\} \rightarrow \text{Hard} \)).

6. APPLICATIONS

As defined in Section 5, a theory of knowledge is extremely general and allows for all of the possible types of competence theory mentioned in Chapter 4. For example, if each rule in \( K \) is required to act independently of the others, a behavioral objectives type theory is obtained. In this case \( J : \mathcal{R} \) includes the S-R pairs associated with a given, usually large, class of tasks to be performed. (Each task defines an equivalence class of \( J : \mathcal{R} \).) \( K \) is a set of rules which accounts for these S-R pairs, one (or more) rule(s) for each task. Hence, such a theory is of the form \( \mathcal{H} = (J : \mathcal{R}, K) \) where \( J : \mathcal{R} \) equals the union of the extensions of the rules in \( K \). Notice that no operations (rules) are allowed on the rules in \( K \). By way of illustration, consider a theory in which \( J : \mathcal{R} \) consists of the tasks of addition, subtraction, multiplication, and division (i.e., the set of number pairs paired with their respective sums, differences, products, or quotients) and \( K \) contains the addition, subtraction, multiplication, and division algorithms (rules).

Allowing the rules in \( K \) to be composed in accounting for \( J : \mathcal{R} \), we get a type of theory which includes generative (phrase structure) grammars. A generative grammar can be characterized by the form \( (J : \mathcal{R}, K, \ast) \), where \( \ast \) is a composition rule which acts on the rules in \( K \) but is not itself in \( K \). For example, we might let \( J : \mathcal{R} \) be any set of pairs whose 

\[ \{x,y\} \rightarrow \text{z} \] denotes the rule described by the program: Start: Go to 1; 1: Do e, go to 2; 2: If \( \{x,y\} \) go to 3, else go to 5; 3: Do r, go to 4; 4: Do d, go to 5; 5: Halt. \( e \) is an atomic encoding rule which inserts stimuli into \( \{x,y\} \) or its complement, \( r \) maps the class \( \{x,y\} \) into \( \text{z} \), and \( d \) decodes \( \text{z} \) (i.e., chooses a particular representative of class \( \text{z} \)).
first element is $S$ (for sentence) and whose second elements are terminal strings (sentences) of the form $a^nb^n$ where $x^n$ ($x = a$ or $b$) means $x$ repeated $n$ times. In this case, an adequate account is provided by letting $K$ contain the two rules $r_1 = S \rightarrow ab$ and $r_2 = S \rightarrow aSb$. The rule $\star$ may operate on $r_1$ and $r_2$ as many times as necessary to generate a rule adequate for generating any particular terminal string. The pair $S \rightarrow aaabbb$, for example, can be generated by the composite rule $r_2 \circ r_2 \circ r_1$ (i.e., apply $r_2$ twice, then apply $r_1$).

Adding transformation rules and allowing them to act on phrase structures, yields a type which includes transformational grammars. In this case, a transformational grammar can be characterized by the form $(\mathcal{L}, R, K, T, t_1, \ldots, t_n, \star^K, \star_T)$ where $T$ is the set $\{t_1, \ldots, t_n\}$ of transformations which act on (sets of) chains of rules in $K$ and $\star^K$ is composition restricted to the rules in and derivable from $K$. Similarly, $\star_T$ acts on $T$ together with $\star^K$. Consider, for example, the theory $(\{ (S, xx) \mid x \text{ is an arbitrary string of } a's \text{ and } b's \}, \{r_1 = S \rightarrow aS, r_2 = S \rightarrow bS\}, \{t = r \rightarrow r' \text{ (where } r \text{ is an arbitrary chain composed of rules } r_1 \text{ and } r_2, \text{ and } r' \text{ has the same input as } r, \text{ and gives the same output except that } S \text{ is replaced by the string which precedes it)}\}, t, \star^K (\text{generalized composition applied to } K), \star_T)$. In providing an account of a given pair in $\mathcal{L}$, $\star_T$ may operate on rules in $T$ and $\star^K$ to produce new (transformational) rules. ($\star^K$ is a transformational rule of sorts since it, too, operates on rules in $K$.) The new rules produced in turn operate on the rules in $K$ to generate new rule chains which when applied to the given input, yield the given output. For example, the pair $S \rightarrow aabaabbabba$ may be accounted for as follows: $\star_T$ produces $\star^K$ followed by $t_\star (\star^K \circ t)$. $\star^K \circ t$ applied to the rules in $K$ yields $r_1$, $r_2 \rightarrow (r_1 \circ r_2 \circ r_1) \rightarrow (r_1 \circ r_2 \circ r_1)'$ where $r_1 \circ r_2 \circ r_1(S) = abaS$ [i.e., $S \rightarrow aS \rightarrow bS \rightarrow abaS$] and $(r_1 \circ r_2 \circ r_1)'(S) = abaaba$ [i.e., $S \rightarrow abaS \rightarrow abaaba$].

In grammars for natural languages, the chains of phrase structure rules in the domains of transformational rules are identical save for the terminal rewriting (phrase structure) rules. Consider, for example, a transformation which acts on phrase structures of the form $N_1V-N_2^{21}$ (e.g., John hit Jim) and generates new phrase structures of the form

```
112 Structural Learning
```

```
N_1   V   N_2
\downarrow \quad \downarrow \quad \downarrow
John  hit  Jim
```

---

21
N\textsubscript{2}\text-sup-V\text-sup-en-by-N\textsubscript{1} (e.g., Jim was hit by John). Each such phrase structure can be thought of as a chain of rewrite rules, identical except for the terminal rewrite rules which output specific words. (Chomsky and Miller (1963, 300-301) disallow transformations on undeveloped phrase markers which correspond here to chains of rewrite rules sans the terminal ones. They argue that this requires duplication of phrase structure (rewrite) rules after transformation.)

We briefly consider an application which involves meaning. In this case, we have

\[
\mathcal{K} = \{ (S,R) \mid S \text{ is a base } b \text{ numeral and } R \text{ is the equivalent base } a \text{ numeral }, K = \{ r_1, r_2, g, *, i \} \}
\]

where (1) \( r_1 \) is a rule for interpreting base \( b \) numerals. Its exact nature is unimportant here (for details see Scandura 1970b; Chapter 7), but its extension has base \( b \) numerals as inputs and can be thought of as having meanings as outputs (i.e., numbers—sets containing appropriate numbers of elements).

(2) \( r_2 \) is a rule for generating base \( a \) numerals given (as input) arbitrary numbers.

(3) \( g \) is a rule which operates on rules for changing numerals in one base (e.g., \( b \)) into numerals in another base (e.g., \( a \)) together with a new pair of base numerals (\( b', a' \)). One element in its domain is \( \langle r_1 \circ r_2, b', a' \rangle \) where \( b' \) and \( a' \) are given by the S-R pair to be accounted for. \( g \) generates new rules for changing (new) numerals in base \( b' \) into numerals in base \( a' \). Notice that \( g \) generates an indefinitely large class of rules.\(^{22}\)

(4) \( * \) is a composition rule and \( i \) is the identity.

Given the rule set \( K \) and the to-be-accounted-for pair S-R (i.e., a pair of numerals in base \( b' \) and \( a' \), respectively) we get

\[
K^2 = K \cup \{ r_1 \circ r_2, * \circ g \}
\]

since \( *(r_1, r_2) = r_1 \circ r_2 \) and \( *(*, g) = * \circ g \)

\[
K^3 = K^2 \cup \{ (r_1 \circ r_2)' \} \cup A
\]

where \( A \) contains the necessary compositions and

\[
g(r_1 \circ r_2, b', a') = (r_1 \circ r_2)' \text{ where } (r_1 \circ r_2)' \text{ is a rule for changing numerals in base } b' \text{ into numerals in base } a'.
\]

This example shows explicitly how \( K^3 \) may depend not only on \( K \) but also on the S-R pair to be accounted for. Also notice that although our theory provides a semantic account, this is not necessary in working with numerals. It would be a simple matter to construct a purely syntactic account which

\(^{22}\) A number of examples of rules like \( g \) can be found in Scandura, Durnin, Ehrenpreis & Luger (1971).
makes no reference to meaning. Presumably, the role of semantics only becomes crucial where $\mathcal{J} : \mathcal{R}$ is sufficiently complex (e.g., as in natural language translation).

7. AXIOMS, COMMENTS, AND CONJECTURES

In general, the nature of PK will depend on the nature of K, and the nature of $\mathcal{J} : \mathcal{R}$ will determine characteristics of any K which accounts for it. For instance, notice that $\mathcal{J} : \mathcal{R}$ may or may not contain pairs including programs that correspond to rules in K. This suggests the following kinds of questions (among others) for future research: (1) Given properties concerning K, determine properties of PK which follow. (2) Given properties of $\mathcal{J} : \mathcal{R}$, determine properties of any K which accounts for it. The more conditions that can be placed on $\mathcal{J} : \mathcal{R}$, of course, the easier in practice it will be to find a suitable K. Finding necessary and sufficient conditions is the ultimate goal. (3) Given two theories $\mathcal{K} = (\mathcal{J} : \mathcal{R}, K, \ldots)$ and $\mathcal{K}' = (\mathcal{J} : \mathcal{R}, K', \ldots)$ such that both K and K' account for $\mathcal{J} : \mathcal{R}$, determine and develop conditions for comparing K and K'.

As described so far, the theory is essentially a schema which may take on any number of forms depending on the particular use to which it is to be put. The axioms for a behavioral objectives type theory, for example, are likely to be far more specific than those required to characterize the more general form of the theory.

Even in the latter case, of course, there are many different forms any particular theory might take. If concerned with the ontogeny of knowledge (e.g., the characterization of knowledge in a way which might reflect its growth from birth), for example, it might seem reasonable to assume that each rule in K has a finite domain. Indeed, in order to insure only the simplest of competencies at birth, one might further require that each rule be atomic and consist entirely of one-instance operating rules (e.g., generalized activity, instincts, reflexes) or decision making capabilities that are two element partitions, one containing one (encoded) stimulus and the other containing the absence of this stimulus. (The latter corresponds to making decisions on the basis of 0's and 1's.) Two such operating rules might be described, "Suck when something soft is in mouth," and "Spit when something soft is in mouth." A related $\mathcal{J} : \mathcal{R}$ might involve distinguishing between situations in which one's stomach is full, and situations where it is not. Composition, probably restricted initially to a small class of rules, might provide an example of a higher order rule. These examples suggest at once the possible importance of generalized tendencies and
"instincts" to the growth of knowledge and the difficulties likely to be involved in identifying all of the crucial ones.\(^{23}\)

If, on the other hand, a theory is concerned with "ongoing" knowledge, then this level of detail makes no sense. People really do know rules that have infinite domains (although they cannot possibly apply the rule in all cases), and a viable theory of knowledge should reflect this. Allowing such rules in \(K\) seems to leave the general scheme unchanged and, furthermore, it is a simple matter to show how rules with infinite domains can be constructed from rules with finite domains by application of higher order rules to lower order ones. The basic idea is simply that the introduction of \(\Delta_m\)'s (nodes) into new rules may lead to "loops" that can be repeated an arbitrary number of times.

Whatever the specific nature of a theory of knowledge, however, it is almost certain to share some common properties. Two that are likely to prove of general value in applications are the following.

**Axiom 1:** For each S-R pair in \(\mathcal{J}:K\) which is (first order) generable from \(PK\), there exists a finite number \(n\) such that the S-R pair is \(n\)th order generable from \(K\).

Axiom 1 disallows "limit" behaviors which cannot be derived in a finite number of steps.

**Axiom 2:** For each class of S-R pairs in \(\mathcal{J}:K\), there is some finite set of rules which accounts for it.

Axiom 2 insures that \(\mathcal{J}:K\) is not pathological and is designed to avoid problems involving diagonalization arguments (e.g., Rogers, 1967), criticisms by certain linguists (cf. Marcus, 1967) interested in natural languages, and the like.

The ontogeny of knowledge is partially described by the following.

**Definition 40:** \(K_1\) is an innate basis for \(K_2\) iff for some finite number \(n\), each rule in \(K_2\) corresponds to some rule in \(K_1^n\) with the same extension.

**Definition 41:** \(K_1\) is a finitary innate basis for \(K_2\) iff \(K_1\) is an

\(^{23}\)It is worth noting in this regard that although Piaget has been centrally concerned with the ontogeny of knowledge, he has for the most part ignored individual differences and concentrated on that which is common to human beings generally. Indeed, in the case of Piaget, \(\mathcal{J}:K\) would correspond roughly to the class of behaviors generable via adult logic. \(K\), then, would correspond to the initial base upon which knowledge grows through the various developmental stages.
innate basis in which each rule has a finite domain.

**Definition 42:** \( K_1 \) has greater simple computing power than \( K_2 \) iff the union of the extensions of the simple rules in \( K_1 \) contains the union of the extensions of the simple rules in \( K_2 \). (Recall that the extensions of simple rules do not involve rules.)

As will become apparent in Chapter 10, the notion of simple computing power has relevance in determining performance where memory is a factor.

**Finitary Conjecture:** Each \( K_2 \) has a finitary innate basis \( K_1 \).

Further, there is a finitary innate basis in which each rule is either an atomic rule with one input or a simple decision rule consisting of a \( dmc \) that partitions a two-element class in half adjoined with two discrete atomic rules with one-element domains.

**Knowledge Potential Conjecture:** If \( K_1 \) is an innate basis for \( K_2 \), then \( \exists m \geq n \) such that \( K_1^m \supseteq K_2^m \).

If true, this latter conjecture could have a number of interesting implications. For example, suppose \( K_1 \) and \( K_2 \) represent the competencies had by two newborn babies, where \( K_1 \) is an innate basis for rule set \( K_2 \), but \( K_2 \) has greater simple computing power than \( K_1 \). Then, it would follow that the baby characterized by \( K_1 \) would have greater knowledge potential but lower initial ability. Thus, behavior that might be accounted for directly (simply) by \( K_2 \) would require \( K_1^n \) for some \( n > 1 \). Coupled with the well-known fact that boys tend to develop more slowly than girls but generally catch and surpass them in certain areas (while presumably the reverse is true in other areas), this observation might provide interesting food for thought by women libertarians. Perhaps the Frenchmen have been right all along. More seriously, this implication could provide a basis for explaining a number of phylogenetic anomalies (e.g., that baby chimps are initially "smarter" than human babies) and, indeed, could place the whole nature-nurture question in new perspective.

---

1 Piaget, of course, does not stop at description. His theory also deals with the ontogeny of the developmental stages. Even here, however, his concern is with the epistemic subject and not with the individual.

2 There are certain advantages of each approach for a theory of behavior. Generative grammars provide a more natural basis for psychological theorizing of the information processing variety, and although I cannot fully justify my claim, I feel that this approach provides a more natural basis for dealing with individual behavior. Conversely, descriptive theories seem to provide a more natural basis for dealing (continued)
Foundations of the Theory of Knowledge

(footnote 2 continued) with group behavior. In fact, an extension of Marcus's (1967, p.2) reasoning suggests that descriptive theories may be a necessity in dealing with certain pathological patterns of behavior.

Note: As it stands, the knowledge potential conjecture is false. A counter-example is obtained by letting

\[ K_1 = \{r_1, r_2, o\} \text{ and } K_2 = \{r_1, r_2, r_1 \text{or}_2, o\} \]

In this case, \( K_1 \) is an innate basis for \( K_2 \) but for all \( m \geq 2 \), there are rules (e.g., \( r_2 \text{or}_2 r_1 \) and \( r_1 \text{or}_2 r_2 \)) in \( K_2^m \) which are not in \( K_1^m \). Nonetheless, given appropriate constraints (e.g., on higher order rules), the conjecture should follow. The study of such conditions will be undertaken at some future time.
The main purpose of this chapter is to show how the ideas discussed in Chapter 4 apply to mathematics. Our basic assumption is that rules are the basic building blocks of all mathematical knowledge and that, if looked at in the right way, all mathematical behavior is rule-governed. More specifically, it is proposed that the mathematical behavior any given subject is potentially capable of at any given stage of learning can be accounted for in terms of a finite set of rules.

Any attempt to deal with the whole of mathematics, of course, would be a hopeless task. What we shall do instead is to indicate some of the kinds of rules that are likely to be involved, and to show how such rules may be combined so as to account for a variety of mathematical behaviors, including what have been called "analogical reasoning" and "creative behavior." Particular emphasis is given to showing how rule sets may grow when (higher order) rules are allowed to generate new rules and to the role such rule sets play in increasing power and parsimony.

To keep the discussion within bounds we make no distinction between rule-governed classes of behavior (computable functions) and procedures. Both are represented by "arrows." Basically, what we do is extend and modify Post's notation for productions to allow rules to act on rules. The use of such notation greatly simplifies the discussion since it obviates the need for describing what are frequently quite complex procedures.

Although hopefully suggestive, the present treatment can in no way be taken as definitive, especially as to detail. Furthermore, we shall emphasize those competencies which are most directly involved in knowing what might be called the content of mathematics. Relatively little is said about the general intellectual skills or processes which necessarily are also
involved in doing most mathematics.\(^1\)

There are basically three different senses of knowing mathematics, and these form the basis for organizing our discussion. The first involves semantics—the internal structure of the objects of mathematical study: embodiments, systems, and families of systems. Most of the examples in Chapter 4 were drawn from this area. In Section 1, we extend these ideas and go into more detail. Section 2 is concerned with syntax—and involves rules which operate within formal systems or directly on them. Section 3 deals with axiomatic theories. The emphasis here is on the various kinds of relationships which exist between syntax and semantics. Special attention is given to the kinds of inference rules involved and to the processes by which such rules are combined. Comments are also made about the rules involved in conjecture-making, constructing counterexamples, and in model theory and metamathematics (proof theory).

Because of the inherent logical nature of mathematics, one might wonder whether an effective procedure exists for generating characterizations (of knowledge) directly from the mathematics itself.\(^2\) I trust that the discussion which follows will make clear that this alone will not be adequate. Mathematical knowledge is sufficiently complex

---

\(^1\) For an introductory discussion of such processes, see Chapter 1 of the author's *Mathematics: Concrete Behavioral Foundations*, New York: Harper & Row, 1971c. Also see Chapter 7 where the question of meaning is discussed. Deductive reasoning is discussed further in Volume II, in Chapter 6.

\(^2\) On cursory examination, for example, it appears that there may be some connection between deducibility of properties and the possibility of generating corresponding rules within a system. For example, notice that the (a) associative and (b) commutative axioms of group theory, respectively, correspond in a fairly "natural" way to the rules

\[(a') (aob)oc = d \rightarrow ao(boc) = d\]

\[(b') aob = c \rightarrow boa = c\]

Similarly, the group property,

\[(c) \forall a, b, c, (aob)oc = (boc)oa\]

corresponds to the rule-governed class

\[(c') (aob)oc = d \rightarrow (boc)oa = d\]

Next, notice that Property (c) may be logically deduced from (a) and (b), and that rule (c') can be generated by composition of rules (a') and (b'). Whether this relationship holds in general, or whether it can be generalized, I do not know since I have not explored the question. In particular, one might ask whether a given property A can be deduced from a set of other properties if and only if a rule corresponding to A can be generated by composition, primitive recursion, and/or particularization (minimalization) from rules corresponding to the other properties.
that any hope of devising a completely systematic and mechanical way of
devising an adequate account (of mathematical knowledge) from a given
mathematical description is undoubtedly unrealistic.

1. SEMANTIC KNOWLEDGE--MATHEMATICAL SYSTEMS AND EMBODIMENTS

Any answer to the question of what it means to know mathematics in an
internal or semantic sense clearly must be compatible with the mathematics
of the situation. But, there is more to it than that. The characteriza-
tion of an algebraic system as an ordered n-tuple, for example, tells us
something about the nature of the object we are studying; but it tells us
very little about what it means to know the system in a detailed behavioral
sense. It is to this task that we now turn.

1.1 Simple Rules and Simple Elements

Semantic knowledge consists in part of simple elements (which do not
act on other elements) and in part of simple rules (which act only on simple
elements).

Computational Rules. In characterizations of semantic knowledge, one
kind of simple rule involves the ability to compute. Consider, for example,
the cyclic three group which was defined in Chapter 3 as the three-tuple

\[ \langle \{A, B, C\}, o = \{A, A \rightarrow A; A, B \rightarrow B; B, A \rightarrow B; A, C \rightarrow C; C, A \rightarrow C; B, C \rightarrow A; C, B \rightarrow A; C, C \rightarrow B; B, B \rightarrow C\}\rangle \]

Knowledge of this system would involve (among other things) rules for
generating each of the "addition facts" associated with the operation o
(e.g. \(A, B \rightarrow B\)). This is certainly possible within the constraints we have
set up because the number of such facts is finite.

The "addition facts," of course, can be generated in a variety of ways.
For example, each fact can be accounted for in terms of a discrete associa-
tion. Another possibility involves repeated application of the generator
element, \(B\). Note that each element of this system can be represented as
a power of \(B\) (i.e., \(A = B^0, B = B^1, C = B^2\)). The ability to "add" there-
fore can be accounted for in terms of the single rule

\[ (1) \quad B^{n \pmod 3}, B^{m \pmod 3} \rightarrow B^{n+m \pmod 3} \]

or, equivalently, as

\[ B^n, B^m \rightarrow B^{n+m} \]

where \(B^x\) represents \(B^0B^1B^2\ldots\) followed by

\[ y_o B o \ldots o y_o \]
an appropriate number of times. For example, application of (1) to \(B, C\) (i.e., \(B^1, B^2\)) gives 
\[B^1, B^2 \to B^0\] (=A)
as is required by the operation \(o\). In effect, the number of rules required in the characterizing rule set is reduced from nine to one.

As impressive as this reduction is, it is still possible to represent knowledge of the "addition facts" in either way. When we turn to a system with an infinite number of elements, however, this is no longer true. In this case, the corresponding basic facts could not possibly be represented as discrete associations. For example, consider the system of whole numbers under addition

\[
\{[0,1, \ldots, n, \ldots], 0, + = \{0,0 \to 0; 0,1 \to 1; 1,0 \to 1; 2,0 \to 2; 1,1 \to 2; 0,2 \to 2; 0,3 \to 3; 1,2 \to 3; 2,1 \to 3; 3,0 \to 3; 4,0 \to 4; 3,1 \to 4; 2,2 \to 4, \ldots\}\}
\]
The operation + consists of a (denumerably) infinite set of facts so that it would be impossible to learn each one as a distinct association.

It would still be possible, however, to account for these facts in terms of generators. In this case, we let \(0 = 0, 1 = s(0), 2 = s^2(0), 3 = s^3(0), \ldots, n = s^n(0)\) so that

\[
n + m = s^n(0) + s^m(0) \to s^{n+m}(0)\]

where \(s^n(0)\) is short for \(\underbrace{s \ldots s}_n (0)\).

The ability to compute, of course, is not necessarily limited to knowing how to add. Among other things, it clearly may involve counterparts of the other arithmetical operations--for example, the ability to determine differences. As viewed herein, knowing the identity (a simple element), say, or the inverse rule, also involves computational ability. The former might be represented \(S \to e\), or \(\rightarrow e\), where \(S\) represents the system and \(e\), the identity. Equivalently, we may simply include \(e\) in the characterizing rule set as a simple element. The inverse rule can be denoted \(a \to a^{-1}\). Determining volumes given three dimensions of solid figures is an example of a ternary operation. Computational ability can also be

---

3 \(n (\text{mod } 3)\) means the remainder obtained in dividing \(n\) by 3. For example, \(14 (\text{mod } 3) = 2\), since \(14 \div 3 = 4\) with remainder 2.

4 The ability to add is necessarily based on a numeration system of one sort or another. That is, reference must be made to \(\text{components}\) of the numerals. Applying the usual addition algorithm to the numerals 45 and 67, for example, involves reference to the corresponding units and tens digits. The numerals are not treated as indivisible wholes but are generated from elements of some finite alphabet. In the case of the base ten numerals, this alphabet is the set \{0, 1, 2, \ldots, 9\}. 
extended to more complex expressions such as \((A \circ B) \circ' (C \oplus B)\) where \(\circ\), \(\circ'\), and \(\oplus\) refer to arbitrary operations and parentheses indicate the order in which the operations are to be carried out. In general, any rule defined on the elements of a system is referred to as a computational rule.

**Simple Elements (Images) in Geometry.** It is instructive to consider the problem of characterizing simple knowledge of geometry because geometries are ordinarily characterized (mathematically) in terms of relations, and not operations. For example, consider the finite geometry 
\[
\langle \{A, B, C, a, b, c\}, P=\{A, B, C\}, L=\{a, b, c\}, R=\{Ab, Ac, Ba, Bc, Ca, Cb\}\rangle
\]
which can be represented pictorially as

![Figure 1](image)

where \(P\), \(L\), and \(R\) are the relations.

Unlike the case with algebraic systems there is no one rule which corresponds more naturally to a given relation than any other. One sense of knowing relation \(R\), for example, would be as a (finite) set of associations
\[
\{A \rightarrow (b,c); B \rightarrow (a,c); C \rightarrow (a,b)\}
\]
or
\[
\{a \rightarrow (B,C); b \rightarrow (A,C); c \rightarrow (A,B)\}
\]
in which the outputs are pairs. Going from pairs of lines to points or pairs of points to lines, of course, would simply require reversing the arrows. There are other possibilities, of course, but these need not concern us here.

Another more complete (yet simple) sense of knowing the geometry pertains to the (above) model of the system itself. Notice that the model denotes a class of observables and is therefore an encoded (simple) element (cf. Chapter 5, Section 5). This simple element is not an indivisible whole, however, but rather an encoded complex corresponding to a stimulus array. In this case, the encoded array is composed of (indivisible) points and lines. Furthermore, we can conceive of any number of simple rules which act on the array to generate properties of the array (i.e., outputs).
It would be a simple matter, for example, to devise a rule which determines the number of lines in the array. Such rules may act on any array in their respective domains so that potentially, at least, they provide a means for generating corresponding properties of other geometries.

It is in this sense, then, that models serve to summarize a good deal of knowledge. The potential of any characterizing rule set depends not only on the presence (in the rule set) of the model itself but also on the presence of rules which act on the model. The ready availability of such rules in most cultures (e.g., the ability to count lines) could account for the fact that many people use "images" in thinking about geometry.

1.2 Relationships Within Systems

It is clear from Chapter 4, of course, that not all rules within a system act directly on elements of the system. Relational rules which act on associations (i.e., degenerate rules) provide a ready example.

In this section we introduce higher order rules which map classes of rules into classes of rules where the individual pairs of rules operate in different systems. In the process we improve on the notation used in Chapter 4. Toward this end, we first consider once again the relational rule between addition and subtraction. In Chapter 4 we used the representation

\[(2) [a, b \triangleright c] \Rightarrow [c, a \leftarrow b]\]

This rule is to be interpreted as taking associations of the form on the left in (2) and permuting them as on the right.

Consider now a closely related but conceptually distinct, degenerate higher order rule which acts on a (nondegenerate) addition rule and generates a corresponding subtraction rule. The notation introduced above would be inadequate for this purpose. The representation gives no hint, for example, as to how c can be constructed from a and b. A simple way to overcome this limitation resides in the fact that addition can be defined in terms of the successor function s which increments by one. In this case, addition can be represented by the rule

\[a, b \rightarrow s^b(a)\]

where \(s^b(a)\) is to be interpreted as s applied to \(a, b\) times. The higher order rule between addition and subtraction, then, might be represented

\[(3) [a, b \rightarrow s^b(a)] \Rightarrow [s^b(a), a \rightarrow b]\]

But, this too has an important limitation. The rule on the right does not tell how to generate differences from arbitrarily given numerals a and b.
As it stands, the subtraction rule works only when the minuend is given in the form \( s^b(a) \).

This limitation can be overcome by letting \(-s\) represent the inverse of \( s\), which subtracts one, so that subtraction can be denoted

\[ a, b \rightarrow (-s)^b(a) \]

Substituting in (3) we get

\[ (4) \quad [a, b \rightarrow s^b(a)] \Rightarrow [s^b(a), a \rightarrow (-s)^a(s^b(a))] \]

There are clearly many other rules of this type. In the system of rational numbers, for example, there are rules which relate addition and multiplication (i.e., as repeated addition), multiplication and division, and subtraction and division. Except for the fact that division by zero is undefined, for example, the (degenerate) rule relating multiplication and division involves nothing basically new.

\[ (5) \quad [a, b \rightarrow f^b(a)] \Rightarrow [f^b(a), a \rightarrow (-f)^a(f^b(a))] \]

for \( a \neq 0 \) where \( f^b(a) \) means the \( b \) fold addition of \( a \). (Notice that \( s \) in (4) corresponds to \( f \) in (5).) The other rules involve rules which act on \( n \)-tuples and are also instructive to consider. The rule between addition and multiplication, for example, may be denoted

\[ (6) \quad [\overbrace{a, a, \ldots, a}^{n} \rightarrow s^a(\ldots(s^a(a))\ldots)] \Rightarrow [\overbrace{s\ldots s(0)}^{n}, a \rightarrow \overbrace{f\ldots f}^{n}(s(0)(a))] \]

The rule on the left of (6) corresponds to \( a + a + \ldots + a \). The rule on the right corresponds to \( n \times a \).

Strictly speaking, we have talked so far only about degenerate higher order rules, whose extensions consist of single rule pairs. It is important to recognize that for each such higher order rule it is possible to devise a corresponding nondegenerate rule, which acts not only within particular systems but also on classes of rules, each of which may be defined within a different system. We may reformulate rule (4), for example, as

\[ (7) \quad [a, b \rightarrow r^b(a)] \Rightarrow [r^b(a), a \rightarrow (-r)^a(r^b(a))] \]

where \( r \) is allowed to vary as well as \( a \) and \( b \). It \((r)\) is a variable, however, only with respect to the higher order rule. It is a constant with respect to any particular lower order rule; and furthermore, the domain of pairs \( \langle a, b \rangle \) depends on the value of \( r \). (For this reason, (7) is not a production in the sense of Post (cf. Minsky, 1967).)

There is no reason, of course, for limiting oneself to first order higher order rules. In multi-operation systems one can frequently find higher order rules which operate on higher order rules. A particularly
simple example can be deduced from rule (4) (the higher order rule between addition and subtraction) and rule (5) (the rule between multiplication and division, except by 0) because they are of essentially the same form. This higher order rule can be represented

\[ (r) \rightarrow (r') \quad a \neq 0 \]

where \((r')\) denotes (7) with \(r\) replaced by \(r'\).

1.3 Semantics and Relationships Between Systems and Embodiments

In this section we consider the question of how to characterize knowledge of relationships between embodiments and systems. We also consider the question of what it means to know the meaning of a system.

Isomorphisms Between Embodiments. Knowing the relationship between two embodiments of the same system clearly involves the corresponding isomorphism. Our job here essentially boils down to specifying the kinds of rules implied. First of all, an isomorphism is a one-to-one map (i.e., rule having an inverse) between the elements of two embodiments. This sounds simple enough, until we realize that an isomorphism is not automatically rule-governed because the number of elements in an embodiment may not be finite. Where the basic set (of the embodiment) is denumerable, however, the elements can be represented (as strings of elements) in terms of a finite set of generators, and a rule corresponding to the one-to-one map (a function) can always be found.\(^5\)

Consider, for example, two embodiments of the system of whole numbers, one involving the usual interpretation (i.e., the set \(\{0,1,2,\ldots\}\) together with the successor operation), and the other, the set \(\{0,2,4,6,\ldots\}\) together with the unary operation "take the next even number." The elements of each embodiment can be represented in terms of a single generator: successor (s) and even (e), respectively. For example, the number 6 in the former embodiment can be represented by ssssss(0), and, in the latter, by eee(0). In this case, the one-to-one map can be represented by the rule

\[ ss\ldots s(0) \rightarrow ee\ldots e(0) \]

which maps arbitrary whole numbers into corresponding even numbers.

Knowing an isomorphism, of course, implies more than just a rule for generating corresponding elements. In particular, knowledge of an operation (rule) in one embodiment automatically implies knowledge of the correspon-

\(^5\)It is worth noting in this regard that isomorphism rules between nondenumerable sets (e.g., the positive reals and logarithms) necessarily involve limit processes.
ding operation in the second. In the example above, we need an (invertible) rule between the *successor* and *even* operations. This can be represented

\[[s^n(0) \rightarrow ss^n(0)] = [e^n(0) \rightarrow ee^n(0)]\]

where \(s^n\) and \(e^n\) denote the strings \(ss \ldots s(0)\) and \(ee \ldots e(0)\), respectively. The relationship between the corresponding addition and subtraction operations, respectively, can be similarly represented

\[[s^n(0), s^m(0) \rightarrow s^{n+m}(0)] = [e^n(0), e^m(0) \rightarrow e^{n+m}(0)]\]

\[[s^n(0), s^m(0) \rightarrow s^{n-m}(0)] = [e^n(0), e^m(0) \rightarrow e^{n-m}(0)]\]

It is important to notice that although these rules act on rules, and are therefore of a higher order, they are degenerate as they stand since each has only one rule in its domain. It should be apparent, however, that such rules can be generalized so as to operate on classes of rules in different embodiments. We return to this possibility below.

Finally, we observe that although the structure preserving properties of isomorphisms refer only to simple rules, existing higher order rules, psychologically speaking, are preserved in exactly the same sense. The relationship between addition and subtraction in the even numbers embodiment, for example, can be generated from the corresponding relationship in the usual interpretation by

\[\delta(n, m \rightarrow n + m)\]

**Semantics.** We now consider the question of what it means to know the semantics of a system. As a first approximation, knowing the semantics of a system may be characterized in terms of any rule (e.g., an association) which maps the system (canonical embodiment) into the denoted class of embodiments (cf. Chapter 3). Notice, however, that the relationship here is between the denoting system and the denoted class, acting as wholes and not between their elements. In particular, if limited to a rule connecting the system to its class of embodiments, a characterization of semantics would be devoid of implications for behavior involving relationships between embodiments.

Clearly, this aspect of semantic knowledge refers to the isomorphisms among the various pairs of embodiments (including canonical ones). Because the number of embodiments of any given system is likely to be indefinitely large, however, complete knowledge of the semantics (i.e., the isomorphisms)
will only be possible where (and if) some finite account can be devised.

With this in mind, we are encouraged to note that the set of isomorphisms together with the operation of composition forms a commutative group. (That is, the system so formed satisfies the closure, associative, commutative, identity, and inverse properties.) This suggests that perhaps a finite set of isomorphism rules, together with the higher order rule of composition, might always exist from which all of the others might be generated. Some further thought, however, suggests that this is highly unlikely. No matter how many isomorphisms a given finite characterization may account for, there is no guarantee that someone might not find another.

In effect, although the meaning of a system may be defined formally (as a class of embodiments), human beings, either individually or collectively, cannot possibly be expected to learn in any complete sense the semantics of an arbitrarily given system. Knowledge of a system in the semantic sense may best be viewed as an ideal which can never be attained in fact.\footnote{Because any finite set of isomorphisms satisfies the associative requirement (where the compositions are defined), the semantics of individual systems may be represented in terms of categories whose objects are embodiments and whose maps are isomorphisms. This form of representation can be used to represent the knowledge of particular individuals relative to some common ideal (cf. Chapter 2, Section 3). The implications (continued)}

\textit{Relationships Between Systems}. Because anything which is isomorphic to a given system is an embodiment of it, rather than a distinct system, we look elsewhere for relationships between systems. Clearly, the other morphisms are likely candidates. Consider, for example, the homomorphism between the system of whole numbers over addition and mod 2 arithmetic (i.e., the system consisting of two elements, the odd (0) and even (E) integers). The systems involved can be characterized, respectively, as

\[
\langle\{0, 1 = s(0), 2 = ss(0), 3 = sss(0), \ldots, n = s^n(0), \ldots\}, \\
0, 1, + = \{s^n(0), s^m(0) \rightarrow s^{n+m}(0) \text{ for all } n, m\}\rangle
\]

and

\[
\{E, O\}, E, 0, \odot = \{(E,E) \rightarrow E; (E,0) \rightarrow 0; (0,E) \rightarrow 0; (0,0) \rightarrow E\}
\]

In this case, the homomorphism maps each even number into the class of evens E and each odd number into the class of odds 0. It can be represented

\[
s^n(0) \rightarrow \begin{cases} 
E \text{ if } n \text{ is even} \\
0 \text{ if } n \text{ is odd}
\end{cases}
\]

As with embodiments, of course, characterizing knowledge of a homomorphism also implies a higher order rule between + and \(\odot\)
As a second example, consider the cyclic-six and cyclic-three groups. To make things concrete, think in terms of the symmetry rotations of a regular hexagon and the symmetry rotations of an equilateral triangle. In this case the homomorphism can be represented

\[
\begin{align*}
&\{0^\circ, 180^\circ\} \rightarrow \{0^\circ\} \\
&\{60^\circ, 240^\circ\} \rightarrow \{120^\circ\} \\
&\{120^\circ, 300^\circ\} \rightarrow \{240^\circ\}
\end{align*}
\]

Knowledge of any other morphism can also be expressed in terms of rules in a similar manner. Thus, an embedding is essentially the same as an isomorphism except that the range is a subsystem.

Another type of relationship between systems is that of recursion or generalization. In this case the relationship is not between the elements in the related systems but between operations (rules) defined on the elements of the systems. Consider, for example, the cyclic three-group

\[\langle \{a^0, a, a^2\}, a, \oplus_a = (a^k, a^\ell) \rightarrow a^{k+\ell} \pmod{3}\rangle\]

and the cyclic five-group

\[\langle \{b^0, b, b^2, b^3, b^4\}, b, \oplus_b = (b^{k'}, b^{\ell'}) \rightarrow b^{k'+\ell'} \pmod{5}\rangle\]

Knowledge of the recursive relationship between these two groups may be represented by the higher order rule

\[\left( a^k, a^\ell \right) \rightarrow a^{k+\ell} \pmod{3} \right) = \left( b^{k'}, b^{\ell'} \rightarrow b^{k'+\ell'} \pmod{5} \right)\]

Like isomorphism, recursion is a two-way relationship. (Homomorphism is not.)

Clearly, two systems may be related in more than one way. For example, the cyclic six-group and the cyclic three-group are related by both recursion and homomorphism. Furthermore, there are many other kinds of relationships between systems. The direct product relationship, for example, provides a way in which two or more systems may be combined to generate a third, as, say, in constructing the dihedral group with two generators of order six from the cyclic three-group and the two-group. The elements of the dihedral six-group can be generated from pairs of elements of the cyclic three- and two-groups by the rule

\[7(continued) for our theory of knowledge are not clear, however, because characterization in this way may tend to mask relationships between the semantics of given systems and other knowledge.

\[8This higher order rule can readily be generalized to represent the recursion relationship between classes of (recursively related) groups. One way to do this is to modify the above higher order rule so that it operates on triples consisting of a rule, a number to serve as the mod value in the generated rule, and a letter (e.g., b) to serve as the new generator element.\]
where \( a \) and \( b \) are elements of the cyclic three- and two-groups, respectively, and \( (a, b) \) is an element in the dihedral six-group.

The quotient relationship is similar in that pairs of systems are involved in generating a third. For example, consider the group of integers \( I \) under addition, together with its subgroup \( S \) of even integers (under addition). The quotient group in this case is the Odd-Even group under \( \oplus \). This relationship can be represented by

\[
(a, S) \rightarrow \begin{cases} E \text{ if } a \in S \\ 0 \text{ if } a \notin S \end{cases}
\]

where \( a \) is an integer. This is not intended to be an exhaustive list of relationships, but hopefully it will be sufficiently suggestive to encourage others to improve on and extend the type of analysis proposed.

**Relationships Between Classes of Systems.** As suggested above, many of the relationships we have identified may easily be extended to operate between classes of systems and between embodiments of classes of systems. In the case of forming products and quotients the process is direct. Indeed, generalization of the characterizing rules above is necessitated by the mathematical essence of these ideas. Forming products corresponds to the rule

\[
(G_1, G_2) \rightarrow G_1 \times G_2
\]

which operates on pairs of systems, say groups, and forms the product group

\[
\langle\{(a_1, a_2) \mid a_1 \in G_1, a_2 \in G_2\}; \langle e_1, e_2 \rangle \rangle \text{ where } e_1 \text{ is the identity of } G_1 \text{ and } e_2 \text{ of } G_2; \langle(a_1, a_2), (b_1, b_2)\rangle \rightarrow \langle a_1 \oplus b_1, a_2 \oplus b_2 \rangle
\]

It is possible, of course, to distinguish between rules that generate the basic set of the product system and those that generate the operations, but there is no advantage here in doing so. Similarly, the process of forming quotients in the case of finite groups can be represented

\[
(G, S) \rightarrow G/S
\]

where \( S \) is a normal subgroup of \( G \) and \( G/S \) is the quotient group

\[
\langle\{aS \mid a \in G\}, S, (a_1S, a_2S) \rightarrow (a_1 \oplus a_2)S \rangle
\]

where the first element of the triple is the basic set, the second is the identity, and the third, the binary operation.

We consider three additional examples. First, observe that the isomorphisms which exist among the embodiments of different systems may
parallel one another in the sense that the elements (and operations) of corresponding embodiments may be related isomorphically by restrictions of essentially the same general rule. The elements in every finite cyclic group, for example, may be embodied either as rotations of a regular polygon or as cyclic permutations of n-tuples. In this case, each isomorphism between rotations and cyclic permutations in any one finite cyclic group corresponds naturally to an isomorphism between corresponding embodiments in every other such group.

We can make this correspondence explicit by representing each correspondence of the isomorphism as a rule

\[
\begin{align*}
360^n/1 &\rightarrow 1 \text{ l-cycle}^n \\
360^n/2 &\rightarrow 2 \text{ l-cycle}^n \\
360^n/3 &\rightarrow 3 \text{ l-cycle}^n \\
&\vdots \\
360^n/m &\rightarrow m \text{ l-cycle}^n \\
&\vdots
\end{align*}
\]

where \(360^n/m\) corresponds to the unit rotation generator of the polygon embodiment with \(m\) sides, which rotates each vertex of the polygon one vertex clockwise, and \(m\) l-cycle, to the unit permutation generator of the \(m\)-tuple embodiment, which permutes each element in the \(m\)-tuple one position to the right (except for the last element which takes the first position). The relationship among the above simple rules can be represented by the higher, order rule

\[
(k, 360^n/m \rightarrow m \text{ l-cycle}^n) \Rightarrow (360^n'/k \rightarrow k \text{ l-cycle}^{n'}^n)
\]

The relationship among these rules may be represented schematically as

\[
\begin{array}{c}
G_1 \\
E_1 \\
E'_1
\end{array}
\begin{array}{c}
\rightarrow
\end{array}
\begin{array}{c}
G_2 \\
0 \\
E_2 \\
0 \\
E'_2
\end{array}
\]

where \(G_1\) and \(G_2\) are arbitrary finite cyclic groups, \(E_1\) and \(E_2\) are rotation embodiments, and \(E'_1\) and \(E'_2\) are permutation embodiments.

Notice, in particular, that abstract systems may be generated in this way just as well as can concrete embodiments. For example, computation rules (operations) in the abstract cyclic three-group can be
generated from computation rules in the rotations embodiment. In effect, computations in an abstract embodiment may be accounted for indirectly in terms of a corresponding computation rule in some concrete embodiment together with an appropriate higher order rule. This observation could help to explain some recent research (Dienes & Jeeves, 1965) in which it was found that adults appear to learn some abstract systems faster than children whereas the reverse is true with other systems. In effect, the adults may have already learned embodiments of some of the systems they were required to learn but not others.

While not directly pertinent, it is also worth noting that abstract systems, when they must be learned independently, may be learned more rapidly than corresponding embodiments. Suppes and his collaborators (e.g., Suppes & Binford, 1965), for example, have found that it takes less time to learn symbolic logic in the abstract than in terms of its meaning and, furthermore, that it takes slightly less time to learn both the symbolic and meaningful versions when the symbolic version is presented first. The reason for this is not entirely clear, but it is quite possible that concrete embodiments contain many more perceptual distractors than symbolic ones and that it takes more time to sort out the irrelevancies.

Another type of relationship follows directly from the fact that every relationship between two systems is reflected between their respective embodiments. Specifically, for each rule which maps the elements and/or operations of one (abstract) system into those of another, there is a class of similar rules between embodiments of these systems. For example, consider the recursion rule

\[
\begin{align*}
(a^k, a^l) &\rightarrow a^{k+l}(\text{mod } n) \\
(b^{k'}, b^{l'}) &\rightarrow b^{k'+l'}(\text{mod } m)
\end{align*}
\]

between two finite cyclic groups, where a and b are the respective generators. The class corresponding to this rule includes the following (in addition to (8)):

\[
\begin{align*}
(r^k, r^l) &\rightarrow r^{k+l}(\text{mod } n) \\
(r'^{k'}, r'^{l'}) &\rightarrow r'^{k'+l'}(\text{mod } m)
\end{align*}
\]

where \(r\) and \(r'\) are unit rotation generators of a regular polygon.

\[
\begin{align*}
(p^k, p^l) &\rightarrow p^{k+l}(\text{mod } n) \\
(p'^{k'}, p'^{l'}) &\rightarrow p'^{k'+l'}(\text{mod } m)
\end{align*}
\]

where \(p\) and \(p'\) are unit permutation generators respectively of \(n\)- and \(m\)-

\[9\]To be precise, this involves a higher order rule between operations in the embodiments. This higher order rule can be represented

\[
\begin{align*}
\left(360^k, 360^l \rightarrow 360^{k+l}\right) &\Rightarrow \left(m \-\text{cycle}^k, m \-\text{cycle}^l \rightarrow m \-\text{cycle}^{k+l}\right)
\end{align*}
\]

where each value of \(m\) corresponds to an operation in a particular system.
tuples.

Since there are indefinitely many rules in this class, they could not possibly constitute an effective account of the related behaviors (i.e., of the union of the extensions of the rules in the class). Due to the close relationship between the rules in this class, however, there is an obvious higher order rule which can be used to generate any particular rule in the class, given any other rule in the class together with the generating elements of the embodiments associated with the desired rule. This higher order rule can be denoted

\[(9) \quad r, p', (8) \Rightarrow \left( r^k, r^l \right) \rightarrow r^{k+l (\text{mod } n)} = \left( p'^k, p'^l \right) \rightarrow p'^{k+l (\text{mod } m)} \]

where (8) is the given rule and \( r \) and \( p' \) are the generating elements.

The illustrative rules preceding (9) are related as shown below:

\[ \begin{array}{ccc} G_1 & \rightarrow & E_1 0 \\ G_2 & \rightarrow & 0 E_2 \\ \downarrow & & \downarrow \\ 0 E_1' & \rightarrow & E_2' \end{array} \]

where \( G_1 \) and \( G_2 \) are finite cyclic groups, \( E_1 \) and \( E_2 \) are rotation embodiments, and \( E_1' \) and \( E_2' \) are permutation embodiments.

Notice that the embodiments corresponding to the rule on the right of (9), unlike in the illustrative rules given above, are not of the same type. The generator \( r \) is a rotation and \( p' \) is a 1-cycle. This is perfectly proper. However, when the embodiments are of the same type, the domain of the higher order rule may naturally be thought of as consisting of pairs rather than triples.

More generally, given a rule relating any two systems and a type of embodiment, together with higher order rule (9) appropriately generalized, it is possible to generate a rule between embodiments of the two systems of the given type. Consider, for example, the following (degenerate) rule between a pair of finite geometries

\[ \begin{array}{ccc} \square & \rightarrow & \triangle \end{array} \]

A rule between another pair of embodiments, in this case, can easily be generated by applying rule (9) to a pair consisting of the above rule and the given type in which, say, the points correspond to people, the lines to committees, and the relation "on" to "being on a committee."

All this suggests that it may not be necessary to learn new embodiments
of every system from scratch. Knowing rule (9) an embodiment of one system, and the relationship of this system to a second automatically confers the ability to generate the corresponding embodiments in the second system.

Our third type involves rules that reflect relationships between classes of pairs of systems, and specifically the higher order relationships that frequently exist among these rules. For example, we consider once again the relation of recursion—in particular, the set of rules between cyclic groups of order n and cyclic groups of order m (cf. rule (8)). This time we need only to introduce a higher order rule which operates on pairs consisting, for example, of rule (8) and a pair of numbers \(\langle n', m'\rangle\), and which generates rules of the form

\[
[(a, a') - a^{k+l}(mod n')] \Rightarrow [(b, b') - b^{k'+l'}(mod m')]
\]

The introduction of this type of higher order rule greatly reduces the number of rules like (8) that must be included in a characterizing rule set. Indeed, rule (8) alone will suffice (as long as a finite account is also included of number pairs).

There is, of course, no reason to stop with relationships between rules that relate highly similar systems of the sort considered. One may conceive, for example, of relationships between certain algebraic and topological families of systems. Although the situation is likely to be more complex, nothing basically new seems to be required. Areas like algebraic topology, for example, might well provide fertile ground for such research. And, then, why not go on to consider relationships between such relationships—and beyond? The level of analysis would seem limited in principle only by the investigator's understanding of the mathematics involved and his ability to translate that understanding into a behaviorally meaningful form.

1.4 Behavioral Implications—Instances of Creative Behavior

In the previous sections a number of different kinds of rules were introduced. Up to this point, however, we have only dealt indirectly with the kinds of behavior that might be generated by given (finite) rule sets.

The main purpose of this section is to show how the addition of just a few basic higher order rules, like composition, makes it possible to generate behaviors which go far beyond those (behaviors) directly associated with the constituent rules. Although none of the examples deal with creative behavior in the strong sense of creating new mathematics, we hope to make a case for the thesis that, in principle at least, creative behavior can be accounted for in this way—specifically, in terms of rule sets in
which rules are allowed to operate on rules.

In each of the following examples, we begin by listing the rule set. Next, we identify new rule-governed classes of behavior which lie beyond the corpus directly associated with the given rules (i.e., the corpus defined by the union of the extensions of the rules). Finally, we show how and why some of these new classes can be generated by combining the given rules and why others cannot be so generated.

Example (1) is particularly simple. The given rules are (a) the ability to add in a given system S and (b) the (higher order) inverse rule which maps arbitrary (binary operation) rules into corresponding inverse rules. To account for subtraction in the given system, we first apply higher order rule (b) to rule (a) to generate the required subtraction rule and then apply this subtraction rule to the stimulus (subtraction problem) in question. This is represented schematically below

\[
\begin{array}{c}
S \\
+ \\
\rightarrow \\
- \\
\rightarrow
\end{array}
\]

where solid arrows refer to given simple rules, double arrows to given higher order rules, and dotted arrows to derived rules.

Suppose now that we modify the initial rule set ever so slightly. Instead of viewing addition as an atomic rule, let us assume that the rule set contains only a finite number of instances (e.g., just the basic addition facts) and, furthermore, that the inverse rule acts directly on these instances viewed as rules in their own right--i.e., as associations. Under these conditions, the ability to account for subtraction instances would be limited precisely to those subtraction instances (e.g., the subtraction facts) corresponding to the known addition instances.

Examples (2) and (3) indicate the importance of the higher order composition rule by which pairs of rules may be combined to form composite ones. We assume throughout that the composition rule applies to every pair of rules such that each output of one serves as part of some input of the other. (The other part of the input may be in the rule set K.)

10 The fact that there are two such rules is unimportant to our discussion.

11 In linguistic accounts of competence, the composition of rules is taken for granted. In the present formulation, we treat composition as just one (important) kind of higher order rule. See Chapter 5, Section 6 for further details.
In Example (2), the rule set also contains (a) the homomorphism from Embodiment $E_1$ of System $S_1$ to Embodiment $E_2$ of System $S_2$, and (b) the isomorphism from $E_2$ to $E_2'$ in System $S_2$. In this case, the composition rule may be applied to rules (a) and (b) to generate the composite rule which operates from $E_1$ in $S_1$ to $E_2'$ in $S_2$. This can be represented

Example (3) is similar except that here the given pair of simple rules (a) the homomorphism from $E_1$ of $S_1$ to $E_2$ of $S_2$ and (c) the isomorphism from $E_1$ of $S_1$ to $E_1'$ of $S_1$ is not in the domain of the composition rule.

If we add higher order rule (9) in Section 1.3, however, we can again derive a rule from $E_1$ of $S_1$ to $E_1'$ of $S_2$. In this case, the composition rule acts on rule (9) and itself, generating the composite rule consisting of rule (9) followed by composition. This composite rule in turn acts first on rule (a) and then on the pair consisting of the output rule ($a'$) and rule (c). The output is the desired rule from $E_1$ to $E_1'$. Higher order rules are omitted in the Figure below for simplicity.
Examples (2) and (3) show, more generally, that "knowing" embodiments of one system, together with appropriate relationships between these embodiments and corresponding embodiments of another system, frequently makes it possible to know the corresponding embodiments of the second system. For example, knowing how to add in arithmetic and knowing the homomorphism between the system (of natural numbers) and mod two (odd-even) arithmetic implies the ability to add in mod two arithmetic. Looked at in this way, we see why it is not necessary to know every embodiment of every system directly. Knowing the embodiments of one system may help one to know corresponding embodiments of another system. Perhaps this is why the mature mathematics student effectively seems to know automatically the intended meanings of many newly presented abstract systems.

One of the most basic activities in which mathematicians engage involves generalizing known results. Example (4) shows how one might account for generalization behavior in the case of semantics. We first formulate the problem in terms of rules; specifically, we equate "knowing a result" with having a rule in the rule set. Generalizing a result, then, means deriving a more general rule from the given one. (One rule is said to be more general than another if its extension properly contains the extension of the other—cf. Scandura, Woodward, & Lee, 1967.) To keep the discussion definite and as simple as possible we consider the familiar rule \( N^2 + N \), which yields the sum of any number series of the form 2 + 4 + 6 + ... + 2N. One possible generalization is the equally familiar rule \( \left( \frac{A+L}{2} \right) N \) for summing arbitrary arithmetic number series. The question is whether or not there is a natural higher order rule by which the latter rule can be generated from the first in a non-trivial way. (Clearly, one could introduce a higher order association between the two.)

To see that the answer is probably no, consider the reverse process, called restriction—specifically, a higher order rule that when applied to \( \left( \frac{A+L}{2} \right) N \) yields \( N^2 + N \). The first step in this higher order rule is to substitute for the first term and the last. Given the rule \( \left( \frac{A+L}{2} \right) N \) and the values \( A=2 \) and \( L=2N \), applying this step yields the rule \( \left( \frac{2+2N}{2} \right) N \). The second step is to simplify, which further transforms the restricted rule into \( N^2 + N \).

We now ask whether or not these steps have natural counterparts that operate in reverse direction. In the case of substitution, the answer seems to be a qualified yes. Thus, given a restricted form like \( \left( \frac{2+2N}{2} \right) N \) in which the "2" in the numerator corresponds to the first term of the
series and "2N" to the last, it seems reasonable to substitute variables for the constants and thereby to generate the more general rule \( \frac{A+L}{2}N \).

It seems less likely, however, that a person might know the specific rewriting rule corresponding to the simplify step (as applied to the rule \( N^2 + N \)).

This observation seems to be borne out in some earlier work (Scandura & Durnin, 1968). Subjects who were given restricted forms of general rules were frequently able to generalize to new instances which were solvable by the general rules. But this very rarely happened where the initially given rules were not obvious restrictions of the required general rule. For example, educated subjects are likely to generalize from \( \frac{2+2N}{2} \) to \( \frac{A+L}{2}N \) but not from \( N^2 + N \) to \( \frac{A+L}{2}N \). Having the advantage of hindsight, let us see how that might happen. We assume that the rule \( \frac{2+2N}{2} \) is known (i.e., in the rule set) together with the higher order generalizing rule, "substitute variables for the given constants." In this case one way to sum the series \( 3 + 7 + 11 + 15 + \ldots + 43 \), for example, would be to apply the Variables Substitution Rule to \( \frac{2+2N}{2}N \) to get \( \frac{A+L}{2}N \), and then to apply the latter.

In principle at least, this type of analysis can be generalized to more complex situations involving generalization. The major problems which are likely to arise in such cases are, in my opinion, more apt to be due to the complexity of the subject matter than to conceptual inadequacy.

In the final three examples, it is not so immediately clear just how the desired behavior can be generated from the given rules.

In Example (5), we assume that the rule set includes: an addition rule in one system, the homomorphism rule between this system and another, and the inverse rule between the class of operations and their inverses.\(^{13}\) In this case, it would be possible to both add and subtract in the derived system, as well as to subtract, in the given one. This can perhaps most easily be seen diagrammatically (Figure A).

In this example, the subtraction rule in \( S_2 \) can be generated by either one of two composite rules, each of which operates on the addition rule in \( S_1 \). One composite rule consists of the inverse rule followed by the

\(^{13}\) We note in passing that the inverse rule does not apply to certain binary operations without modification. Multiplication in the system of rational numbers is restricted in this way for the simple reason that division by zero is not defined. (If one deleted zero, the inverse rule would apply perfectly well.) It would be quite possible, of course, for a person to know an inverse operator of this sort but not know its mathematical limitations. Indeed, this is exactly what happens all too frequently in the mathematics classroom.
homomorphism. The other consists of the same rules in reverse order. In order to make explicit how the composites are constructed we include in the rule set a higher order composition rule.

In Example (6), we assume that the rule set contains rules for adding in $E_1$ of $S_1$, an isomorphism from $E_2$ to $E'_2$ in $S_2$, a homomorphism from $E_1$ to $E_2$, and a higher order composition rule which (among other things) applies to morphisms. This set of rules allows for a variety of behaviors and, in particular, makes it possible to add in $E'_2$. According to our analysis, the composition rule, when applied to the given homomorphism and isomorphism,
yields the composite morphism. The composite morphism, in turn, when applied to the addition rule in \( E_1 \), yields addition in \( E'_2 \). The latter, of course, can be applied to any addition problem in \( E'_2 \) to obtain the solution.

As a final example (7), suppose that the rule set contains the generalization rule \( g'_2 \) from \( S'_2 \) to \( S'_3 \), where \( S'_2 \) is a cyclic two-group and \( S'_3 \) is a cyclic three-group. Assume also that the rule set contains (a) a higher order rule \( F: g_n \rightarrow g_{n+1} \) where \( g_n \) is a generalization rule from \( S_n \) to \( S_{n+1} \) for \( n \geq 2 \) and (b) a composition rule, called compose, which applies to all rules in the rule set (including compose itself).

Given these rules, the question is whether \( S_n \) can be generated from \( S'_2 \). What is needed is a rule which applies to \( S'_2 \) and yields \( S_n \). We can imagine one, the composite rule

\[
(10) \quad g_2 \circ g_3 \circ g_4 \circ \ldots \circ g_{n-1}
\]

But, we do not know whether (10) can be derived from our rule set. Again, however, we can imagine a higher order rule which would work

\[
(11) \quad \underbrace{F \circ \text{compose} \circ F \circ \text{compose} \circ \ldots \circ F \circ \text{compose}}_{(n-1) \text{ times}}
\]

This rule works as follows: \( F \) is applied to \( g_2 \) to yield \( g_3 \), compose is applied to \( g_2 \) and \( g_3 \) to yield \( g_2 \circ g_3 \). Similarly, \( F \) applied to \( g_3 \) yields \( g_4 \), compose applied to \( g_2 \circ g_3 \) and \( g_4 \) yields \( g_2 \circ g_3 \circ g_4 \) and so on until (10) is obtained.\(^{14}\) Rule (11), in turn, can be generated by applying compose to \( F \) and compose, giving \( F \circ \text{compose} \), and repeatedly thereafter \( n-3 \) more times to the successive outputs and either \( F \) or compose.

Throughout this discussion, we have assumed that a rule set can account for a given behavior (S-R pair) if and only if it contains a rule for generating that behavior or if such a rule can be derived either directly or indirectly from rules in the rule set. We were not very explicit about what it means for a rule to be derivable. (However, see Chapter 5, Section 5.) We simply showed by example how higher order rules may be used to generate (new) lower order rules which may, in turn, be used to generate still lower order rules until an appropriate rule is derived.

In example (7), we implicitly illustrated a heuristic procedure for determining whether or not a given rule set is adequate for a given derivation. Stated more generally, the procedure goes as follows:

\(^{14}\)The same recursive pattern is involved in computing factorials. To compute \( 5! \), for example, we generate the next number 4, combine it with 5 (by multiplication) to get 20, generate the next number 3 from 4, combine it (3) with 20 to get 60, generate 2 from 3, and so on until 1 is combined with the composite 120 (\( = 5 \cdot 4 \cdot 3 \cdot 2 \)).
(a) Test to see if there is a rule in the rule set which generates the desired behavior directly. If so, we are finished.

(b) If not, try to construct a rule that does generate the desired behavior, using rules that are apt to be generable from the rule set. This must be accomplished primarily on heuristic grounds.

(c) Test to see if there is a rule in the rule set which can be applied to other rules in the set to generate the constructed rule.

(d) If not, try to construct a rule that will generate the initially constructed rule.

This process is continued level by level until we succeed or are convinced of failure, in which case we can start over again if we feel that there are other alternatives worth exploring.\(^\text{15}\)

Two final comments are in order. First, the notion of derivability is closely related to the mechanisms of human behavior. Derivability via a rule set corresponds directly to problem solving ability in the idealized theory of Chapters 7 thru 9. In this theory, learning does not (indeed cannot) take place in a vacuum as conditioning theorists would have it. Rather, it is proposed that what a subject is capable of learning in any given situation depends directly on what he already knows. The stimulus situation simply provides the occasion for learning and performance.

Our final comment concerns the need for drawing out relationships between the proposed theory and recursion theory. For example, two conjectures in need of formal proof or disproof are: (1) If a rule is derivable from a given (finite) rule set, then the function this rule generates is equivalent to a partial recursive function. (2) There is some (finite) rule set such that every partial recursive function can be generated by a rule derivable from the rule set.

2. SYNTAXIC KNOWLEDGE—FORMAL SYSTEMS

In this section, we ask what it means to know formal systems. The basic problem once again is to identify the kinds of behavior which knowing a formal system makes possible and to show how such behavior might be accounted for.

Because of the way formal systems are defined, the answers to these questions are relatively straightforward. First of all, knowing a formal system implies the ability to generate the theorems (i.e., strings of

\(^{15}\)In theory, there is no finite upper bound on the number of levels that may be involved. In interpreting this description, it should also be noted that any particular rule may operate at more than one level in the course of a given derivation.
symbols) of the system from the axioms. Accounting for such behavior is a relatively simple task since all theorems may be generated by repeated application of the given inference rules (productions) of the system. The question of alternative rule sets does not arise, at least not in the sense described earlier, because the set of inference rules from which all generating procedures are constructed is fixed once and for all for any given system—by definition.

Furthermore, any number of effective schemes (higher order rules) exist for making sure that each possible combination of the given inference rules is eventually tried, thus insuring that all possible theorems can be generated. As an example, consider the scheme which first selects each inference rule singly, then all pairs, triples, and so on. Combinatorial schemes of this sort act on inference rules and, hence, play a higher order role in generating theorems. That is, they generate sequences of inference rules (procedures), which in turn generate theorems from the axioms.16

Such schemes may apply to classes of formal systems (possibly all), and correspond (and are computationally equivalent) to what in the artificial intelligence literature (cf. Nilsson, 1971) are called breadth first methods. There is, however, an important conceptual difference. In breadth first methods, as in most (if not all) search methods in artificial intelligence, rules are applied as they are selected. In the case of formal systems, each inference rule would be selected and applied individually. If the goal theorem is reached, the search stops. Otherwise, the inference rules are again selected and applied individually, this time to each of the outputs of the preceding step. The process is continued until the theorem is proved or some preset depth level is reached. In our scheme, by way of contrast, the rules are combined first (singly, in pairs, triples, etc.) before they are applied.

Unfortunately, proving theorems using this scheme, or any other non-heuristic method for that matter, would take so long and rapidly become

---

16 Knowing recursive systems implies, in addition to being able to generate proofs of all theorems, the ability to decide whether or not arbitrarily given well-formed statements of the language are theorems (i.e., can be generated by the inference rules). Thus, any complete characterization of the underlying knowledge would necessarily have to also include a characterization of the decision procedure itself. Of course, since the decision procedure in the case of recursive systems is effective (i.e., algorithmic) by definition, this causes no theoretical difficulties. The statement logic, for example, is recursive and the familiar truth table procedure for determining the truth or falsity of given statements can easily be expressed as an algorithm.
so tedious that one would hardly want to call the behavior mathematical. Consider, for example, the formal system defined by

Axiom 1: \( a \)

Rule 1: \( a \rightarrow aa \)

Rule 3: \( b \rightarrow bb \)

Axiom 2: \( b \)

Rule 2: \( a \rightarrow ab \)

Rule 4: \( b \rightarrow ba \)

where each rule is presumed to apply in any context. In this case, to prove the theorem \( bbba \), one could simply apply the rules in order to each of the axioms, first singly, then in pairs, and so on as indicated above. This would eventually yield the theorem but the procedure would clearly be the long way around.

To make the task more interesting, it is necessary to impose constraints on the way proofs are constructed. For example, instead of trying all possible combinations of inference rules beginning with single rules, why not begin, say, with combinations including one less inference rule (three) than the number of symbols in the target theorem (four)? Because all of the inference rules in the system are constructive (i.e., add an \( a \) or \( b \)), this would undoubtedly be a more efficient procedure but probably still not come very close to the sort of behavior that might be elicited by a skilled mathematician.

A somewhat more realistic approach might be to test each rule in turn to see if it generates outputs that are consistent with the desired theorem. Rule 4 is consistent, for example, because it outputs elements of the form \( xbay \), where \( x \) and \( y \) are strings of \( a \)'s and \( b \)'s. The same is true of Rule 3. Rules 1 and 2, however, are not consistent because they output elements of the forms \( xaay \) and \( xaby \), and hence are involved in generating the theorem \( bbba \). Next, one of the consistent rules is selected. We then test the rules again; this time the selected rule must apply to the outputs of the first rule and generate outputs that are consistent with the theorem. The next step is to form the composition of the first and second rules selected. This process is repeated a third time before the (three-component) composite inference rule is applied. If the output matches the theorem, we are finished; otherwise, we must start over again.

It may be worth noting in passing that this heuristic method corresponds in the artificial intelligence literature to means-ends analysis (Ernst & Newell, 1969). In the latter, the heuristic (higher order) rule would go something as follows: (1) Choose an Axiom (say, \( b \)), (2) Try in turn the various inference rules which apply (Rules 3 or 4), (3) Compare...
each result (bb, ba) and see if it has any more a's or b's than the target bbba. (Remember the rule set is strictly constructive.) If not, proceed with another inference rule which applies; otherwise, start over. (4) Apply the next inference rule... and so on. Again, the basic difference between this and our higher order rule is that in our case the rules are not applied one by one, as the rules are selected, but only after a derivation has been completed. Any number of distinct derivations may be required, of course, before a goal is reached in our way.

Whether or not one feels that any of these heuristics comes close to how a skilled mathematician might go at the problem is not important. The really essential point is that there exist different ways of generating proofs. Some may amount to nothing more than simply brute force trial and error. Others may be truly elegant; and most will lie somewhere in between.

Evaluation of different characterizations, in the case of formal systems, is particularly tricky because the typically used criteria of power and parsimony can be misleading unless properly interpreted. If we restrict ourselves solely to formal systems (with a finite number of inference rules), for example, we could obtain a highly parsimonious and powerful characterization by simply listing the respective inference rules and the higher order breadth first scheme described above for combining inference rules in all possible ways.

As pointed out above, however, this sort of characterization captures little of the elegance one normally likes to attribute to knowing mathematics. In addition, the larger the rule set, the more inefficient systematic trial and error becomes.

As we shall see in the next section, the situation becomes even more complex in axiomatic mathematics, where there are fewer ground rules for choosing inference rules. In this case, furthermore, no general scheme exists for proving all true statements (about all axiomatic theories) no matter how much time one is allowed. To the extent that proving theorems in formal systems is to parallel proving theorems in axiomatic mathematics, reliance on heuristic methods for constructing proofs becomes absolutely essential. (Unfortunately, it is not at all clear how relevant to human behavior are such heuristics as means-ends analysis and resolution (cf. Nilsson, 1971) which have been used in artificial intelligence—claims to the contrary notwithstanding.)

Furthermore, when we consider what it means to know more than one formal system, we immediately raise the question of possible relationships between the systems, and thereby the possibility of reducing the number of
inference rules (productions) required. Productions in different systems may, for example, be of the same general form even though they may operate on quite different kinds of strings; one might involve, say, the English alphabet and another, the Arabic numerals. Thus, the following productions are distinct but of the same general form.

\[ x A y \rightarrow x B y \quad \text{where } x, y \text{ are arbitrary strings of } A's \text{ and } B's \]

\[ x' 0 y' \rightarrow x' 1 y' \quad \text{where } x', y' \text{ are arbitrary strings of } 0's \text{ and } 1's \]

In this case, one can conceive of a higher order rule relating them.

\[ x A y \rightarrow x B y \Rightarrow x' 0 y' \rightarrow x' 1 y' \]

By the same argument that we have made many times before, knowledge of the above two productions and the higher order rule may be characterized in terms of the higher order rule and just the first production. The increase in power obtained by including higher order rules is perhaps more clearly appreciated by recalling that such rules may be generalized so that they map classes of productions into classes of productions. This might arise, for example, where we have two reasonably well-defined and parallel classes of formal systems.

3. AXIOMATICS—RELATIONSHIPS BETWEEN SEMANTIC AND SYNTACTIC KNOWLEDGE

In Section 2, we saw that syntactic knowledge is concerned with rules which act directly on signs, without reference to the entities, relations, or operations these signs denote. The syntax of even natural languages, for example, at least to a first approximation, may be characterized as a certain type of formal system, in which there are a number of distinct syntactic categories (e.g., noun, verb; term, predicate) and a number of grammatical rules which act on elements in these categories.\(^\text{17}\) Semantic knowledge, on the other hand, is concerned with "meaning"—in mathematics, with the internal workings of a system,\(^\text{18}\) or of families of systems. More generally, semantic knowledge is concerned with rules that apply to objects, relations, and/or operations denoted by signs. In this sense, the behavior involved in fixing an automobile is assumed to be rule-governed just as much as problem solving behavior in arithmetic.

Although the latter type of knowledge has long characterized school and even much of college mathematics, relatively little attention is typically given to formal systems outside courses in logic and computer science. Nonetheless, the sort of knowledge most closely associated with

---

\(^{17}\)See Engeler (1968, 3-12) for a formal treatment of grammars and elementary languages.

\(^{18}\)Throughout this section, the term "system" may be thought of as a canonical embodiment.
contemporary mathematics, indeed all knowledge, is neither wholly semantic nor wholly syntactic but rather involves close relationships between both kinds. In particular, in addition to syntactic and semantic rules, there are also, in the present view, processing rules which assign meanings to signs and composite signs, and, conversely, rules for constructing descriptions of given meanings. As one might suspect, it is not a simple matter to identify such rules, but some beginnings in this direction have been made (cf. Winograd, 1971; also see Section 6 in Chapter 7 on expository learning).

In this regard, we just mention that simple interpretative rules apply directly to statements, while higher order interpretative rules apply to lower order (e.g., simple) interpretative rules. Thus, specific rules might assign "meaning" to such terms as "group," "If a, then b," "subset," "x is a function of y," and so on. Higher order rules, then, might serve to integrate simple rules so as to make it possible to generate the meanings of more complex statements outside the domains of the simple rules. In this case, one might use higher order rules as before to generate new interpretative rules which then are applied to the statements in question. In mathematics, my guess is that we intuitively recognize the existence of such rules when we make such remarks as, "The only prerequisite to this book is some degree of mathematical maturity."19

For present purposes we shall simply assume the existence of those processing rules needed to pass between statements in a language under consideration and the underlying meanings. This makes it possible in principle to account for a given behavior by operating at either the semantic or syntactic levels, and using the necessary processing rules for converting statements into meanings and/or vice versa. Consider, for example, the statements "Joey is running home" and "Janie is running home," together with the associated composite sentence "Joey and Janie are running home." The composite sentence in this case may be generated from the two simple ones in either of two ways. Thus, we might apply a simple trans-

---

19In practice, it is possible to test for the ability to describe an idea by first making sure that the subject "has the idea" and then seeing whether he can describe it in a satisfactory way. Conversely, describing a new idea and then seeing whether the subject can perform in accordance with it (the idea) might be used to test for interpretative ability. For example, if a subject is able to interpret the statement, "If S is a subgroup of G, then the order of S divides the order of G," then given a cyclic-three and a cyclic-five group the subject ought to be able to respond correctly if asked whether the first is a subgroup of the second.

See Chapter 8 and Section 13 of Chapter 7 where our basic approach to assessing behavior potential is described in detail.
formational rule (cf. Chomsky, 1957) to the simple sentences viewed as P-markers. We might also accomplish the same thing by first applying rules for interpreting each of the simple sentences. In this first case, for example, the output might be a scene in which Joey is actually running home, and similarly in the second case, the output might be a scene in which Janie is running home. We may then operate on these two meanings at the semantic level to generate a composite picture of both Janie and Joey in the act of running home. The final step would be to generate a sentence which describes this composite act. It would be a simple matter to construct any number of similar examples (as long as we do not have to detail the rules); witness the transformations from "Jeanne is running" to "Jeanne ran," and from "Julie did her homework" to "Did Julie do her homework?" In each case, the interrelationship between semantic and syntactic knowledge may be represented schematically

![Syntactic Rules](image1)

![Semantic Rules](image2)

![Meaning Rules](image3)

Not only can behavior be generated at both the semantic and syntactic levels, then, but according to this analysis, syntactic behavior can be generated via semantic knowledge, and vice versa. Viewed in this manner, a number of problems currently under intensive investigation make no sense as formulated and must be modified. Any attempt to find out whether or not thought in general can take place without words, for example, is a non-question. In most situations, most people can do both. On the other hand, of course, it is quite reasonable to ask such questions as what kinds of thought are less likely to take place without words, or which modes of thought are preferred by or within the capacity of most subjects at particular developmental levels.

Axiomatic mathematics is concerned with a particular kind of relationship between syntactics and semantics—in particular, with logical relationships among properties of mathematical systems at the semantic level, and with their mirror images at the syntactic level. In this section, we shall attempt to identify and describe some of the kinds of rules needed to characterize knowledge of axiomatic mathematics. We emphasize those competencies that appear necessary in proving given theorems. In addition, a few speculations are made concerning the way counterexamples may be constructed and how conjectures may be generated in the first place.
Unfortunately, with the exception of Polya's (1962, 1965) insightful but atheoretic discussions of problem solving and the current work on computer simulation and artificial intelligence (e.g., Ernst & Newell, 1969; Feigenbaum & Feldman, 1963; Minsky, 1968), there is little to report. Rather than attempt to review this work here, our aim is to show how the problem may be formulated within the general theoretical framework proposed.

Given a corpus of theorems (and/or proofs), our problem boils down to devising a finite set of rules from which each of the given theorems (proofs) can be generated from the axioms and previously proved theorems. If the corpus is finite, a set of discrete associations between, say, the theorems themselves and their respective proofs might suffice. As many aspiring tenth-grade geometers know, however, this approach works only so well. True, one can memorize a reasonably large number of different proofs, but what happens when one is presented, say on a test, with a new theorem one has never seen before?

Clearly, some people can succeed on such problems, and the question is, why? What kinds of rules make this possible? We might be tempted to treat axiomatic mathematics in the same way as formal systems, but even our brief discussion of the relationship between syntactic and semantic rules convinces us that this would be an inadequate solution. Some appeal to semantics seems indispensable in proving theorems if we are to reflect human knowledge. One of the first things that the mature problem solver typically does in proving theorems, for example, is to figure out what the given theorem statement means—that is, to interpret it. (The verb "figure out" was used purposefully since it is quite compatible with the interpretative mechanism proposed in Chapter 7.)

The intended meaning of the theorem "If two distinct lines A and B are parallel to M, then A is parallel to B," for instance, may be visualized directly.

---

The related but more difficult question of whether or not it is even possible to devise a syntactic rule set which is fully adequate is left for later comment. We simply mention in this regard that Gödel's Incompleteness Theorem suggests that any attempt to ignore semantics may not be feasible with axiomatic theories, that are at least as complex as number theory. The problem is that no one axiom set provides a sufficient basis from which to generate all possible well-formed formulas or their negations. That is, some such formulas are undecidable. (Henkin's extended completeness theorem, of course, tells us that given the axioms of any axiomatic theory, the logical axioms and rules of inference of the predicate logic do in fact provide an adequate basis for generating all of the theorems of that theory.) The question of what truth is, appears basically to be a semantic question, and any attempt to reduce it entirely to syntax would surely be in error.
The key relationships indicated by the theorem are indicated by arrows, with $P_1$ and $P_2$ corresponding to different parts of the premises, and $C$, to the conclusion. Reasoning at the semantic level, then, may be thought of as a succession of transformations of meaning from the premises to the conclusion. Such transformations may range in generality from association-like to universally applicable semantic inference rules. The former correspond to so-called rote proofs, however, and are not considered below.

In order to construct a proof, of course, reasoning that is done at the semantic level must be converted back into statements at the syntactic level. A fully adequate rule set, therefore, would necessarily have to contain rules for interpreting statements and for describing meanings as well as for reasoning at both the semantic and syntactic levels.\(^\text{21}\)

Although recognizing the importance of such processing rules, we shall concentrate here on those kinds of rules which involve reasoning and which are therefore more unique to proof construction. Section 3.1 deals with inference rules—specifically semantic inference rules which are so essential in doing axiomatic mathematics. Higher order rules for combining semantic inference rules are considered in Section 3.2. Section 3.3 sketches some of the kinds of rules involved in constructing counterexamples and shows how conjectures may be devised in the first place. Finally, some brief comments are made about metamathematics and model theory.

3.1 The Nature of (Semantic) Inference Rules

The general nature of the stimuli, responses, and operations involved in what we shall call semantic inference rules is perhaps best understood by analogy to formal systems.

In formal systems, theorems (strings of symbols) are generated by application of formal inference rules to axioms (other strings) and/or

\(^{21}\)As important as they are, specific discussion of interpreting and describing rules is postponed until Chapter 7, where relevant learning and performance mechanisms are also introduced. The learner is also referred to Chapter 1 in the author's *Mathematics: Concrete Behavioral Foundations*, where the basic processes involved in mathematical behavior are classified, described, and illustrated in some detail.
previously proved theorems.\textsuperscript{22} These formal axioms and theorems correspond to properties of mathematical systems. The string "For all $x$, there exists an $x'$ such that $x \circ x' = e$" for example, corresponds to the inverse property.\textsuperscript{23} Such properties define families of systems, namely those families whose systems each have the specified property. For example, the "closure" and "associative" properties each define a distinct family of systems; one can find systems which are closed, but which do not satisfy the associative property, and vice versa. Thus, the system of integers over subtraction is closed, but not associative (e.g., $(^+4 - ^+3) - ^+1 = 0$ whereas $^+4 - (^+3 - ^+1) = ^+2$). The conjunction of these two properties defines the family of systems known as semi-groups.

Properties, of course, define classes (families) not only with respect to mathematical systems, but also with respect to entities of any sort whatsoever. Given arbitrary number series, for example, the property of having a common difference of two between adjacent terms defines the class \{a + (a+2) + (a+4) + ... + [a + (n-1)2] | a and n are positive integers}. This suggests a close relationship between properties of mathematical systems and perceptual (encoding) rules (cf. Chapter 5, Section 2). Specifically, properties of systems are nothing more than outputs of perceptual rules applied to given systems (or parts thereof) acting as effective stimuli. To the extent, then, that formal systems reflect what actually goes on at the semantic level, properties (or equivalently, families) of systems are internalized (encoded) stimuli and responses.

We now ask what rules of inference correspond to at the semantic level. In formal systems, rules of inference map one or more strings called axioms (and/or previously proved theorems) into other strings called theorems. Since these strings may be made to correspond to properties in axiomatic theories, and since each such property defines a family of systems, we see that at the semantic level, syntactic rules of inference correspond to transformations between families of systems, considered as encoded stimuli and responses.

\textsuperscript{22}More specifically, these rules apply to equivalence classes of strings.
\textsuperscript{23}Such strings of symbols, of course, are nothing more than predicates in formal systems, which are constructed from atomic predicates by use of one or more logical connectives (not, and, or, implies, etc.). Atomic predicates, in turn, correspond roughly to (one or more) rules operating within particular systems. (For this purpose, we allow atomic predicates to be bound by quantifiers, but no other logical connectives.) Thus, for example, the binary predicate "For all $x$, there exists an $x'$ such that $x \circ x' = e$" corresponds to the rule "$x \leadsto x'$." The unary predicate "There exists an $e$ such that $x \circ e = x$" corresponds to the rule "$\rightarrow e$" (i.e., the generation of $e$ from the system itself). A nullary (0-ary) relation corresponds to a property of the system taken as a whole.
Semantic rules of inference are unique in an important sense—they apply universally. That is, each rule of inference may be applied in any stimulus situation whatever, in the sense that the stimulus necessarily has certain properties to which the inference rule applies. All other rules have a limited range of applicability. Trying to form the "passive form" of a number series or to "add" a series of nonsense syllables, for example, really would be nonsense.

Some idea of the way semantic inference rules operate can be obtained by considering some familiar syntactic rules of inference. Modus ponens provides a simple illustration. Suppose that the statements "If G is a finite group, and S is a subgroup of G, then the order of S divides the order of G" and "G is a finite group and S is a subgroup of G" are properties of one family of systems (actually, a family of pairs of systems) and that "If a function is continuous over a closed interval of the real line, then it is uniformly continuous" and "The function is continuous over a closed interval of the real line" are properties of another family. Then, application of modus ponens tells us that "The order of S divides the order of G" and "The function is uniformly continuous" are also properties of their respective families. The corresponding premises and conclusions are quite different, but the rule of inference by which they are related is identical. In effect, the transformation involving groups and that involving continuous functions, while superficially quite different, are instances of exactly the same inference rule (modus ponens).

Unfortunately, the parallel between syntactic inference rules and
transformations between families of systems is not perfect. It is incomplete in several senses. Certain (syntactic) inference rules, for example, are not normally given in a form which directly parallels the underlying semantics. Perhaps the primary example of this involves instantiation, an inference rule that is usually denoted

\[ \forall x \ P(x) \vdash P(a) \]

At the semantic level, this inference rule simply says that any property had by all elements of a system is necessarily had by each one. More specifically, the underlying semantic transformation maps families defined by pairs of properties of the types "For all x, P(x)" and "x = a" into families defined by "P(a)." Thus, for example, if a system has the properties "For all x and y, x + y = y + x" and "x = a," then it may also be expected to have the property "For all y, a + y = y + a." Similarly, given the properties "For all x, \( \sin^2 x + \cos^2 x = 1 \)" and "x = a" it follows that "\( \sin^2 a + \cos^2 a = 1 \)." This inference rule, of course, yields a different result for each particular value of x. The upshot of all this is that in order to parallel the underlying semantics exactly, instantiation at the syntactic level would have to be denoted

\[ \forall x \ P(x) \vdash x = a \vdash P(a) \]

Substitution for equals may be similarly treated. In this regard, an unnamed critic (and friend) proposed that "The futility of trying to think of rules of inference (even) as functions is already evident once one considers substitution of equals." But I think a little thought will convince anyone that this inference rule maps pairs of the form

\[ P(x_1, x_2, \ldots, x_n), x_1 = a_1 \]

into elements of the form

\[ P(x_1, \ldots, a_1, \ldots, x_n) \]

Thus, the initial form defines a class of functionally distinct stimuli (e.g., \( P(x_1, x_2, \ldots, x_n), x_1 = a_1; P(x_1, x_2, \ldots, x_n), x_2 = a_2; \ldots \)); so it is not surprising that one can generate any number of different responses (e.g., \( P(a_1, x_2, \ldots, x_n); P(x_1, a_2, \ldots, x_n); \ldots \)).

The parallel is also incomplete because there are transformations between families of systems (properties) that do not correspond to what most logicians call (syntactic) inference rules. Consider, for example, the two properties "A" and "B," each of which defines a distinct family of systems. Any system that has both properties will necessarily also
have certain other properties, in particular the conjunctive property "A and B." (Other such properties include "A or B," "A implies B," and so on.) The family of semi-groups, for example, has the conjunctive property of "closure and associativity." In this case, one can clearly envision a transformation between the families defined by "closure" and "associativity" and the family of semi-groups (i.e., the family defined by "closure and associativity"). But this transformation between families does not correspond to what logicians typically call an inference rule. It rather corresponds to a rule for constructing new predicates from old ones. In effect, inference rules as normally defined do not exhaust the possible kinds of logical transformation rules at the semantic level.

There are also forms of logical reasoning at the semantic level that do not involve transformations between families. Such rules correspond to what at the syntactic level Corcoran (see Chapter 6, Volume II) calls "lengthening rules." Lengthening rules operate on logical arguments, or chains of reasoning, and generate statements. Corcoran calls such reasoning "subsidiary reasoning" because it involves reasoning from subsidiary proofs to steps in main proofs.27

To see what such rules correspond to at the semantic level, consider the case where the reasoning goes from a derivation of "B" from "A" (denoted \( A \vdash B \)) to the conditional "A implies B." (Together this may be denoted \( (A \vdash B) \vdash A \Rightarrow B \)) The derivation \( A \vdash B \), of course, corresponds to a chain of transformations (i.e., an argument) of the sort described above, together possibly with other subsidiary proof rules. (Alternatively, it means that the family defined by A also has property B or, equivalently, that the family defined by A is included in the family defined by B.) The conclusion \( A \Rightarrow B \), of course, just defines a family of systems, a family which includes every system except those which satisfy A but not B. Hence, the semantic counterpart of a lengthening rule goes from chains of transformations (e.g., from instances of a composite of semantic inference rules) to families of systems.

Indirect proofs involve another example of subsidiary reasoning. In this case, the chains of transformations correspond to arguments which show that the premises, together with the negations of what are to be proved (the conclusions), lead to contradictions. The outputs of the

27Corcoran's discussion is an attempt to synthesize work on natural deductive systems (cf. Anderson & Johnstone, 1962; Kalish & Montague, 1964) and competence theories, with particular reference to linguistics.
lengthening rule are theorems (i.e., families of systems defined by theorems). A third example involves proof by induction. In this case, we show that some property holds when \( n = 1 \) and that whenever it holds for \( n \), it must also hold for \( n + 1 \). From this pair of arguments, we conclude that the property holds for all \( n \).

The main thrust of Corcoran's work, as I see it, is to characterize the deductive process at the syntactic level in a way which parallels the underlying semantics—and which reflects human reasoning. By way of introduction to Corcoran's chapters and in order to show how his discussion of the problem fits within the framework discussed here, some additional comment is in order. Corcoran's use of the term "correct" in reference to suppositional theories of proof stems directly from his attempt to devise at the syntactic level a characterization of deductive reasoning which directly parallels semantic intuition. Insofar as accounting for logical reasoning per se, many different logical theories are formally equivalent to the sort of suppositional theories about which Corcoran is talking. "Correctness" should not be interpreted to mean that suppositional theories are somehow more adequate, say, for doing proof theory. Most logicians feel that the reverse is true. It is generally simpler to work with linear theories (See Chapter 6, Volume II). However, in order to evaluate alternative characterizations of logical reasoning (at either the syntactic or the semantic levels), one must look beyond the "truth" or logical adequacy of the characterization. Evaluating various alternatives will of necessity be relative to what the investigator has in mind. The psychologist, for example, will no doubt be concerned with behavioral relevance and the underlying semantics. The logician will probably want an efficient characterization which readily lends itself to formal manipulation. The mathematician, on the other hand, will undoubtedly be most concerned with efficient techniques for checking particularly difficult parts of proofs.

Insofar as behavior is concerned, there is no reason to believe that the inference rules subjects actually learn are necessarily identical to those which logicians use in their formulations. Consider, for example, a generalized form of modus ponens. This procedure applies to premises like \( A \supset (B \supset (C \supset D)) \), \( A \), \( B \), and \( C \), and generates conclusions of type \( D \). Basically, the procedure involves the simple inference rule modus ponens together with appropriate branching instructions. Thus, the first step in "computing" the above instance involves applying modus ponens to \( A \supset (B \supset (C \supset D)) \) and \( A \), giving \( B \supset (C \supset D) \). The next step involves \( B \) and \( B \supset (C \supset D) \), giving \( C \supset D \). The final step involves \( C \) and \( C \supset D \), giving the conclusion \( D \). The branching
decisions are contingent on finding an appropriate single letter (A, B, or C) which allows the indicated detachment.

There is no reason why complex logical rules might not be used in generating theorems in exactly the same way as simple ones. Indeed, single logical rules may be sufficient for generating proofs of fairly complex theorems. Furthermore, many logical procedures, even reasonably complex ones, are apt to be common to a number of different theories. The number of more or less unique procedures in any particular theory is likely (according to the present view) to be relatively small. Hence, assuming prior mastery of most standard logical procedures, a skilled mathematician may gain mastery of a new theory in relatively short order by concentrating on those procedures associated with some of the more novel proofs in the theory.

Logical procedures bear a close relationship to the proofs people write. Specifically, they correspond to transitions between steps and from arguments to steps (in subsidiary reasoning). Consider, for example, the following proof that \( \sqrt{2} \) is irrational.

1. (a) Assume \( \sqrt{2} \) is rational.
   
   (b) Then \( \sqrt{2} = \frac{p}{q} \) where \( p \) and \( q \) are relatively prime (i.e., do not have a common factor other than 1).
   
   (c) And \( 2q^2 = p^2 \).
   
   (d) Hence, \( p \) and \( q \) are not relatively prime.
   
   (e) Therefore, \( \sqrt{2} \) is irrational.

The transition from step (1a) to step (1b) corresponds to a logical procedure consisting of the inference rule instantiation followed by modus ponens. Application of instantiation to

(a') For all \( r \) if \( r \) is a (positive) rational, then there exist natural numbers \( p \) and \( q \) such that \( r = \frac{p}{q} \) where \( p \) and \( q \) are relatively prime.

yields

(a'') If \( \sqrt{2} \) is rational, then there exist natural numbers \( p \) and \( q \) such that \( \sqrt{2} = \frac{p}{q} \) where \( p \) and \( q \) are relatively prime.

Application of modus ponens to (1a) and (1a''), then, gives (1b).\(^{28}\) Step (1c), in turn, can be generated in a similar manner from (1b) by using certain basic theorems about rational numbers. Step (1d) also follows directly from previous steps. In each case, the logical procedures correspond to transitions from given properties (e.g., assumptions and theorems).

---

\(^{28}\) Notice that the rule set which accounts for this proof must contain certain properties (e.g., (1a')) in addition to the logical procedures.
to conclusions (also properties). Step (le), however, does not follow by application of logical procedures to properties, but rather by application of lengthening rules to the argument (an instance of a compound logical procedure) from (la) to (ld).

Each such transition in a given proof roughly parallels a logical procedure that the subject who wrote the proof knows. In fact, the transitions in a proof so viewed may be thought of as syntactic counterparts of semantic operations. (The ability "to write" the various steps, of course, depends on certain requisite processing (descriptive) skills. 29)

The relationship between logical procedures and transitions in proofs is nowhere near perfect, however. Proofs may be written at various levels of detail, as long as the transitions correspond to logical procedures known to the subject. Each level of detail corresponds to a particular set of logical procedures. For example, the sort of proof a mathematician might construct to convince himself that a statement is true might be quite different from the type of proof he might write to convince others. In the former case, the transitions are far more likely to correspond to the more complex of the logical procedures he knows. These are the kinds of deductive leaps of which the mathematician is capable. The transitions in the latter case, however, are generally designed to correspond to those logical procedures which are most characteristic of those who are to read the proofs. (This last statement may perhaps be a bit too strong since the transitions typically used depend to some extent on tradition. In working with formal systems, furthermore, transitions are limited to small sets of specified inference rules.)

Of course, if the sole concern of an investigator is to come up with an efficient account of a given corpus of theorems and/or proofs, it is irrelevant whether or not the various proof transitions correspond to the most complex logical procedures known to the person who wrote them. In this case, the investigator can simply act "as if" the logical procedures corresponding to the transitions apparent in the proofs are the most complex ones available. In testing theoretical assumptions concerning human behavior (cf. Chapters 7&8), however, it is absolutely necessary to know precisely which (relevant) logical procedures the subject in question actually knows. 30

29 For a more complete discussion of processing skills, see my Mathematics: Concrete Behavioral Foundations. New York: Harper & Row, 1971c. The reverse process of interpretation (i.e., going from syntax to meaning) is considered in Chapter 7.

30 George Lowerre and I (1972, Structural Learning Report No. 65) have just completed a study involving critical reading in which we (continued)
3.2 On Generating New Logical Procedures

In order to generate most proofs, indeed to successfully account for complex deductive reasoning of any sort, a rule set must contain more than just discrete rules of inference, or even a large number of relatively complex logical procedures. According to the present analysis, the rule set must also contain higher order rules which make it possible to generate new logical procedures.

To see this, we first observe that there is an important difference between making (generating) inferences and generating logical procedures. The former involves generating theorems by application of logical procedures to axioms and/or previously generated theorems. The latter corresponds to generating logical procedures, which in turn can be used to generate theorems from axioms and/or other theorems. Where an appropriate logical procedure exists at the semantic level, constructing a proof of a given theorem simply amounts to transforming the underlying semantics into syntactic form. Since we assume throughout Section 3 that appropriate interpretation and description rules are available where needed, the difference between proofs and logical procedures for generating theorems may safely be suppressed for purposes of discussion.

The important point here is that in general there will be theorems whose proofs do not correspond to any logical procedure in a characterizing rule set, no matter how powerful and diverse that rule set might be. Such proofs frequently correspond to new logical procedures that must be derived from other rules in the rule set.

An Intrinsic Limitation of Rule Sets (Optional). Even if one allows given rule sets to grow, they will still be inadequate in a more subtle sense, in view of Gödel's Incompleteness Theorem. The gist of this theorem, as it relates to the present situation, is that there will always be conjectures about systems in any given sufficiently complex family of systems, that can neither be proved nor disproved within any formal system used to characterize the family. Equivalently, it is impossible to construct logical procedures, which generate such theorems, using only the logical procedures (i.e., rules of inference) and axioms associated with the formal system. Furthermore, this will still be true even where the system (more accurately, rule set) is allowed to grow by generating new logical
procedures. (This follows directly because anything that can be generated by the new procedures can be generated without them.) An important result of this observation is that no one finite set of logical procedures (and axioms) provides a sufficient basis for generating proofs of all supportable statements about most families of systems.

This suggests that in order to allow proofs for all true statements in axiomatic mathematics, it may be necessary to introduce an indeterminate number of different formal characterizations (i.e., rule sets consisting of axioms and/or logical rules) for any given family. To see this we first recall that there are some theorems that have so far resisted attempts to prove them without using the so-called axiom of choice (cf. Rosser, 1953). In its essentials, one form of the axiom of choice states simply that: Every set (including infinite ones) can be well-ordered—or, equivalently, that any number of choices from the set is permissible. The majority of mathematicians rebel at the idea that more than a finite number, or at most a denumerably infinite number, of such choices might be made. If each choice took some finite amount of time, no matter how small, there would simply not be time to complete the selections (cf. Rosser, 1953, 490-492). In spite of its repugnance, the axiom of choice remains an essential tool in mathematical research.

Now, the axiom of choice is not an axiom (property) in the sense that it defines some specific family of systems. It is rather a tautology—a property schema (i.e., a family of properties), which may be viewed as a logical procedure in its own right. Specifically, as with all tautologies, the axiom of choice may be thought of as a rule for inferring new properties from given ones. In this case, the rule may be thought of as mapping the universal property of "having a basic set," which all systems do have, into the equally universal property of having an order on that set. More generally, the semantic inference rules corresponding to tautologies may transform properties of specific families into universal properties. The tautology $P \lor \neg P$ provides a simple example. It involves inferring $P \lor \neg P$ from any given property $P$.

There are several other tautologies (logical procedures) commonly used by mathematicians, which are equally controversial. Respectively, these correspond to (Stoll, 1963, 116)

(a) Zermelo's axiom of choice: For every set $X$, there exists a function $f$ on the collection of nonempty subsets $A$ of $X$ into $X$, such that for each $A$, $f(A)$ is a member of $A$.

(b) Hausdorff's maximal principle: Every partially ordered set includes a maximal chain (i.e., a chain which is not a proper subset
of any other chain).

(c) Zorn's lemma: Every nonempty partially ordered set in which each chain has an upper bound contains a maximal element.

Moreover, these statements have all been shown to be equivalent both to each other and to the simple form of the axiom of choice given above (Stoll, 1963, 116).

This fact, however, in no way implies that still other (perhaps even more controversial) logical procedures might not also be introduced. To be sure, introducing such procedures as basically new inference rules could have equally as profound an effect on mathematics as did the recognition over 100 years ago that the parallel postulate in geometry was simply an assumption. (This recognition, of course, led to the development of axiomatics as we know it today.) Perhaps it is just this sort of re-conceptualization which may be needed to remove such old mathematical scars such as Fermat's famous last theorem and Goldbach's conjecture. Neither has been proved within the context of formal logic and number theory, and it is certainly possible that the problem may be due to their basic inadequacy (as forecast by Gödel's Incompleteness Theorem). If so, the only way to prove such conjectures may be to introduce entirely new kinds of inference rules (or tautologies).

This would be very much like assuming that, given any conjectured property about a family, which has some specified kind of support, there exists some formal system which describes the given family up to an equivalence in which the conjecture can be proved. The underlying conception of truth in this case, of course, might differ greatly from that normally held.31

Nature of Higher Order Rules in Axiomatics. In this section, we consider a few of the more obvious characteristics of higher order rules involved in proving theorems. Toward this end we recall first that a given theorem can be proved only if the characterizing rule set contains appro-

31It is interesting to entertain the possibility, although it is by no means certain, that introducing new kinds of inference rules could greatly increase the domain of mathematical study. In particular, there are many phenomena in the social sciences, philosophy, the humanities, and everyday life, which while appearing to defy mathematization, using traditional mathematical logic, might well yield to nonstandard formulations. Human emotionality, esthetics, and other forms of art, for example, could conceivably benefit from such study. I would like to be more specific since the general problem could either be a terribly important one or sheer nonsense, but unfortunately I cannot. Perhaps the problem is just felt and not real, but, in any case, I hope that someone will be willing to investigate the matter sufficiently to set the record straight.
appropriate logical procedures together with higher order rules for combining such procedures in the required way.

As with formal systems, logical procedures and corresponding proofs can be constructed in axiomatic mathematics via systematic trial and error, means-ends analysis, resolution, or any other variant of the general heuristics used in artificial intelligence. Owing to the greater complexity and variety of the logical procedures involved in axiomatics, however, broadly applicable higher order methods of this sort are likely to be even less useful than with formal systems.

Trial and error, for example, is not likely to be very useful in generating proofs in axiomatic theories, because the number of possibilities would be so great as to be impractical to enumerate, even with high-speed computers, let alone human beings. More important, no one seriously believes that mathematicians prove most theorems by trial and error (although it is quite possible that a certain amount of this may be involved).

Although complete specification of higher order rules is beyond the scope of this discussion, some feeling for what is involved may be obtained from examples. Specifically, what are the higher order rules like, and how do they operate in proving theorems?

One type of higher order rule that may be used in constructing proofs is closely associated with the general heuristic "Work backward from the unknown." In this case, proofs (logical procedures) are constructed by first selecting an inference rule which yields the conclusion, and then trying to derive a logical procedure (by using this or other higher order rules) which yields the input of the inference rule.

The widely used technique of proving theorems indirectly by assuming that the conclusion is false provides a special case of this heuristic. Here, the first step in constructing a proof amounts to selecting what might be called the contrapositive lengthening rule by which the theorem \( A \Rightarrow B \) is inferred from a proof of \(~A\) from \(~B\).\(^{32}\) For illustrative purposes, consider the following theorem and proof.

(2) Theorem: The \( \sqrt{2} \) is irrational.

Proof: (a) Assume \( \sqrt{2} \) is rational (i.e., there exists a \( p \) and a \( q \) such that \( \sqrt{2} = \frac{p}{q} \) where \( p \) and \( q \) are relatively prime)

(b) Then \( 2q^2 = p^2 \)

\(^{32}\)This logical procedure may be broken down into two simple inference rules by inserting an intermediate step. First, the contrapositive \(~B \Rightarrow ~A\) is deduced from an argument showing how \(~A\) follows from \(~B\) by a lengthening rule. The theorem \( A \Rightarrow B \), then, is inferred from \(~B \Rightarrow ~A\).
(c) Now $2$ is a factor of $p$ (since it is a factor of $p^2$) and $4$ is a factor of $p^2$.

(d) Hence $2q^2 = 4r (= p^2)$ or $q^2 = 2r$ and $q$ has a factor of $2$.

(e) CONTRADICTION ([(a) and (c) and (d)]), so $p$ and $q$ are not relatively prime.

(f) Therefore, $\sqrt{2}$ is irrational.

Although there are any number of ways in which this proof could be generated, we consider only one, a way which is compatible with our form of analysis and which, at the same time, involves an indirect proof. In order to simplify the analysis we assume that logical procedures are available which parallel the various allowable transitions between the steps of the proof. Specifically, we assume that logical procedures are available by which (2b) can be generated from (2a), (2c) from (2b), (2d) from (2b) and (2c), and (2e) from (2a), (2c), and (2d). The availability of the lengthening rule from the argument from (2a) to (2e) to step (2f) is also assumed.

In this case, the "work backward" rule forms the composite of two higher order rules. First, it selects a rule for selecting the lengthening rule which goes from arguments of the form $\sim B$ to $\sim A$, denoted $\sim B \vdash \sim A$ (e.g., (2a)$\vdash$(2e)), to statements of the form $A \supset B$ (e.g., (2f) where $A$ is understood to include all axioms of the real number system). Then, the "work backward" rule adds on to the beginning a rule for combining logical procedures, such as those involved in going between steps (2a) and (2e). We shall say little about this type of rule here. (It would be easy to construct a higher order rule which will do the job, but identifying a rule that accurately reflects human knowledge is another matter. As indicated in the section below, this requires analysis of a class of similar proofs.)

The composite of the two higher order rules, in turn, can be used first to combine the logical procedures involved in going between steps (2a) and (2e), and second, to add on the lengthening rule which applies to arguments like (2a) $\vdash$ (2e). The output of the composite higher order rule, then, is a logical procedure which corresponds directly to the proof itself.

There are, of course, many other general proof-generating rules. Proof by cases and proof by induction are two of the more common. Even direct proofs fall into this pattern. (Given a theorem $A \supset B$, the general idea in a direct proof is to generate an argument from $A$ to $B$ ($A \vdash B$) and, from this, to infer $A \supset B$.)

Consider a second proof. Generating this proof involves both the direct proof paradigm and proof by cases.
(3) Theorem: In the system of integers \( I \), the subset of odd integers is closed under multiplication.

Proof: Case 1 \((n, m \geq 1)\): (a) Let \( n = 2x + 1, m = 2y + 1 \) where \( x, y \) are both integers greater than or equal to 0 \((x, y \geq 0)\).

(b) Then, \( n \cdot m = (2x + 1)(2y + 1) = 4xy + 2x + 2y + 1 \) which is odd.

Case 2 \((n, m \leq 1)\): (c) Let \( n = -2x + 1, m = -2y + 1 \) where \( x, y \geq 0 \)

(d) Then, \( n \cdot m = 4xy - 2x - 2y + 1 \) which is odd.

Case 3 \((n \geq 1, m \leq 1)\): (e) Let \( n = 2x + 1, m = -2y + 1 \) where \( x, y \geq 0 \)

(f) Then, \( n \cdot m = -4xy + 2x - 2y + 1 \) which is odd

(g) Q.E.D.

Again, there are any number of ways in which this proof may be constructed; the following analysis seems to be one of the more reasonable ones. The first step in one higher order rule is only implicit in the proof and might be to decide upon a proof by cases—that is, to select a (semantic) lengthening rule by which the theorem (Step (9)) may be inferred from arguments for the three cases. The next step might be to combine logical procedures corresponding to the three cases. The logical procedures used in proving the three cases are practically identical. In each case, the steps follow directly from simple properties of the integers. The result, essentially, is a new logical procedure which generates the conclusion from the premises. The final step of the higher order rule, then, is to compose this new logical procedure with the lengthening rule so that the former is applied first.

In the next section, we deal with the problem of how to identify higher order rules. In particular, we shall outline a quasi-systematic way in which to go about this most difficult task.

On Devising Rule Sets that Account for Proofs. In devising rule sets, the central problem is not one of generating individual proofs. That can be done in any number of ways. The problem is how to devise rule sets that provide viable and efficient accounts of given classes of proofs. Although there is no entirely mechanical way to accomplish this, we shall
Perhaps the most obvious thing to do in analyzing a given corpus is to look for parallels among the proofs. These parallels may be of various types. In the simplest case, precisely the same logical procedure may be involved in generating two or more theorems in the corpus. (Logical procedures, recall, correspond directly to proofs.) Because of the great variety of possible proofs, however, such theorems will generally be few and far between, except where the very simplest inference rules are involved—cf. the illustration in Section 3.1 concerning modus ponens.

A somewhat more typical, but still reasonably rare, situation arises where the general form of two or more proofs is the same but where the logical procedures corresponding to the particular transitions between steps may be different.

Such parallels often provide good justification for introducing higher order rules into a characterization. For example, consider the following proofs. The first is a simple variant of (2) in which transitions between (2b), (2c), and (2d) are collapsed to form one more complex transition from (2b) to (2d). The second (4) has precisely the same form.

(4) Theorem: A triangle T can have at most one right angle.
Proof: (a) Assume T (a triangle with 180°) has more than one right angle.
(b) Then the sum of the angles of T = (90° + 90°) + X°, where X° > 0° is the measure of the third angle X.
(d) Hence T has more than 180°.
(e) Contradiction ((a) and (d)).
(f) Therefore, T can have at most one right angle.

There are three main things to note here. First, not only do the steps in the two proofs parallel one another but so do the individual transitions. In both cases, the transitions go from (a) to (b), (b) to (d), and (a) and (d) to (e). Second, logical procedures corresponding to these proofs may be generated by the same higher order rule. (This higher order rule would involve both the indirect proof rule and a rule for combining the transitions between steps (a) and (e).) Third, although the corresponding transitions are entirely parallel, they may not (and generally speaking, will not) correspond to the same logical procedures. The transition between steps (4b) and (4d), for example, might correspond to instantiation applied to (4b) and
(5) For all numbers \( a \) and \( b \), if \( b > 0 \), then \( a + b > a \), which is equivalent to letting \( a = \theta = 180^\circ \) and \( b = x^\circ \). Substitution (instantiation), then, gives \( 180^\circ + x^\circ > 180^\circ \) (or, step (4d)). The corresponding transition in proof (2) (between steps (2b) and (2d)) cannot reasonably be accounted for so simply.

Although perfect parallels of the sort just given are not particularly easy to find (although they can more easily be constructed), parallelisms between parts of different proofs are far more common. The indirect proof rule, for example, is used in a wide variety of proofs—as is the direct proof rule, and even proof by cases and proof by induction. Such rules may even be embedded within one another in various classes of proof.

There is, of course, no reason for parallels to stop at the level of proofs. For one thing, there is good reason to believe that important parallels may exist between developments (i.e., sequences of proofs) of possibly different theories. In introducing group theory, linear algebra, and many other kinds of algebraic system, for example, it is traditional to start with the weakest possible axioms. The process usually begins by proving certain standard (and trivial) consequences of these axioms. Then, further definitions are usually introduced, and the development proceeds in earnest.

Perhaps the simplest and clearest example, however, involves the so-called duality principle. Recall that the duality principle refers to the fact that in certain theories, the proof of each theorem can be generated mechanically from the proof of its dual. In the algebra of sets, for example, the dual of any statement can be generated by replacing "U" by "\( \cap \)" or "\( \cap \)" by "\( \cup \)" and vice versa. Thus for example, given a proof of

\[
(6) \quad A \cup B = (\overline{A} \cap \overline{B})
\]

then the proof of

\[
(7) \quad A \cap B = (\overline{A} \cup \overline{B})
\]
can be generated by applying the duality rule to the statements in the proof of (6). Clearly, the availability of such a rule would have obvious implications for the way in which the algebra of sets might be developed. The dual of any theorem would most likely be listed as a corollary. Other versions of the duality principle hold in the statement logic, projective geometry, and category theory.

More important, there are cases where the duality principle holds between what may appear on the surface to be distinct theories. The relationship between Boolean algebra and the statement logic provides a parti-
cularly simple example. Here, statements in logic correspond in one-to-one fashion with algebraic expressions in the sense that "\( \lor \)", "\( \land \)" and "\( \neg \)" correspond, respectively, to "union," "intersection," and "complement" in Boolean algebra. (For an elementary discussion of this relationship, the reader is referred to my *Mathematics: Concrete Behavioral Foundations*.)

The identification of higher order rules which enter into the organization of developments would certainly go a long way toward increasing understanding of just what mathematical knowledge consists of. Any viable account of proof-making associated with Boolean Algebra and the Statement Logic, for example, might reasonably include rules for generating proofs in the Statement Logic, given proofs of the corresponding dual theorems in Boolean Algebra.  

Important parallels may also exist beyond the level of developments. For example, specific relationships (parallels) may be detected between different classes of developments. Algebraic topology might well provide fertile ground for such study. In this case, the goal would be to pin down some of the higher order parallels which apparently exist between different algebraic theories, on the one hand, and different topological theories, on the other.

3.3 Conjecture-Making, Constructing Counterexamples, Proof and Model Theory

There is more involved in axiomatics, of course, than constructing proofs of given theorems. The way conjectures are arrived at in the first place, for example, may have little directly to do with logical reasoning. Furthermore, nothing has been said about the rules involved in constructing

---

33In view of the obvious complexities involved in accounting for proof generating behavior, one might reasonably expect the number of different kinds of rules required to be truly astronomical. And this may be so. My guess, however, is that the number of basic kinds of rules will turn out to be much smaller.

The kinds of rules needed could bear some relationship to those used to characterize partial recursive functions. Specifically, no matter how complex the behavior, it may be sufficient to consider just five kinds of rules.

1. Rules for selecting available rules.
2. Rules for modifying other rules (or inputs) in some common way. Modus ponens is one such rule.
3. Rules for forming compositions of rules—that is, for stringing rules together so that the outputs of the first serve as inputs for the second.
4. Rules for generalizing other rules—i.e., for substituting variables for constants which enter into the rule.
5. Rules for restricting other rules. Such rules involve substituting constants for variables and correspond to minimalization (cf. Chapter 2).
counterexamples—in showing that given conjectures are false.

**Conjecture-Making.** One important source of new conjectures is old theorems. In particular, one of the first things the mathematician tries to do after proving a given theorem is to see if it can be generalized. This process is a basic one, and corresponds to rules for replacing constants in given theorems with variables. The familiar Bolzano-Weierstrass Theorem of Elementary Analysis (cf. Apostol, 1957, 43 & 49) provides a simple example.

If a bounded set \( S \) in \( E_1 \) (the real line) contains infinitely many points, then there exists at least one point in \( E_1 \) which is an accumulation (limit) point of \( S \).

As it stands, this theorem applies only to the real line, but it can easily be generalized by simply replacing \( E_1 \) (the real line) in the statement by \( E^n \) (\( n \)-dimensional space). This gives a generalized version of the Bolzano-Weierstrass Theorem, one which can be proved in precisely the same way as for the one-dimensional case, by appropriately generalizing the corresponding definitions for \( E_1 \).

"Conjectural" processes of this sort, then, may take place at a strictly syntactic level by operating on statements themselves. Other conjectures in axiomatic mathematics, however, are primarily of a semantic sort.

Corollaries, for example, are typically generated from theorems by simple logical procedures that might well be explicitly included in a characterizing rule set. Consider the well-known theorem of LaGrange (cf. Birkhoff & MacLane, 1953).

The order of a finite group \( G \) is a multiple of the order of every one of its subgroups.

The following corollaries can be generated as indicated.

- The order of every element of a finite group \( G \) is a divisor of the order of \( G \).

This corollary follows logically from the theorem and the fact (another theorem) that each element of \( G \) generates a cyclic subgroup, whose order is by definition the order of the element.

- Every finite group of prime order has no nontrivial subgroups.

This is a simple restriction of LaGrange's Theorem.

- Every group \( G \) of prime order \( p \) is cyclic.

This follows from the theorem and the fact that each element \( e \neq e \) of a group generates a cyclic subgroup.
No noncyclic group $G$ is of prime order.

This follows directly from the preceding Corollary by simple inference. Generating proofs of such corollaries, of course, also involves the application of description rules to the respective logical procedures.)

Perhaps most conjectures in axiomatics, however, are arrived at by induction on particular systems. The basic type of rule involved is again one of generalization, only this time the inputs are classes of systems defined by properties and the outputs (conjectures) are larger classes defined by more general properties. It is worth noting in this regard that there are two fundamentally different kinds of conjectures, and each requires a different sort of induction. Some conjectures involve properties of systems, taken as wholes, without reference to individual elements. LaGrange's Theorem is of this general sort if one thinks in terms of pairs of systems (i.e., group-subgroup pairs) rather than just single systems. In this case, the primary induction takes place over distinct pairs of systems of various orders. Theorems of the statement logic also fall in this category.

In most cases, however, conjectures tend to involve reference to internal aspects of the systems involved. The Bolzano-Weierstrass Theorem is of this type. Here, the main reference is to what is essentially a single system, the real line. The primary induction in this case takes place not over different systems but over bounded infinite subsets of the system. Goldbach's conjecture, that every even integer greater than 2 is the sum of exactly two primes, is also of this sort, as is Fermat's famous last theorem.

Many conjectures, of course, may require induction, both within systems and across them. Most theorems in group theory, where the models (systems) are not all isomorphic, are of this type.

This brief discussion of conjecture-making is obviously nowhere near complete. Hopefully, however, it may provide a useful starting point for more detailed research in the area.

*Constructing Counterexamples.* To complete our overview of the deductive process, we need also to say something about how conjectures are shown to be false. The basic process, of course, involves constructing *counterexamples*—systems in which the premises of the conjecture hold, but not the conclusion.

This process is in some sense the opposite of generalization and

---

34The basic learning processes involved in this type of activity are discussed at length in Chapter 7, in the section on learning by discovery.
corresponds to what we have called restriction rules (cf. Chapter 5, Section 4). At the syntactic level, restriction rules impose constraints on one or more variables in the statement of a conjecture. For example, consider conjecture

(A) A continuous real-valued function over a bounded interval is uniformly continuous.

A counterexample can be obtained by requiring that the continuous real-valued function increase asymptotically over a bounded interval which is open. Similarly, a counterexample to conjecture

(B) Every finite group is cyclic

can be constructed by requiring that the finite group have more than one generating element.

The corresponding semantic operations involve fixing values of one or more irrelevant dimensions of systems satisfying the premises of given conjectures. Consider, for example, the continuous function $x^2$ over the closed interval $[a, b]$. This model is uniformly continuous and therefore satisfies both the premises and the conclusion of conjecture A. In addition, this model has any number of specific properties which are irrelevant insofar as the conjecture is concerned. Thus, for example, there is no reason why the interval must be closed; according to the conjecture, it need only be bounded above and below. Similarly, one could choose any other function over such a domain, as long as it is continuous. If, in this case, we make the interval open $(a, b)$ and keep the same function $x^2$, we still have a valid model of the conjecture. But, if we introduce the continuous function $(b - a)/(x - a)$, which is unbounded as $x$ approaches $a$, this no longer holds since this function is not uniformly continuous over $(a, b)$. With respect to conjecture B, we need only to break away from considering finite groups with single generators. As soon as we do this, there are any number of finite groups which are not cyclic.

In summary, the two main points are simply that: (a) Counterexamples may be constructed by rules operating at either the syntactic or the semantic levels and (b) Such rules amount to imposing restrictions on the variables involved and, as such, are in some sense opposite to generalization rules.

There is more that could be said about these processes, particularly when viewed in the perspective of the learning mechanisms discussed in Chapter 7, but we shall have to leave this as a nontrivial exercise for the interested reader. Attention is called particularly to Section 5 of Chapter 7 on discovery learning.
Proof and Model Theory. Those working in mathematical foundations are frequently concerned with such questions as whether or not it is even possible to find a proof or a counterexample for a class of conjectures. Questions of this sort cannot be answered within any given axiomatic theory, but rather involve working in a meta-theory.

Metamathematics and model theory are both informal axiomatic theories in which formalizations play the semantic role and where only relatively simple inference rules are allowed. In metamathematics, each statement defines a class of formal systems, each of which has the property denoted by the statement. This is just another way of saying that the meaning of a statement is a class of formal systems. The statement "The set of sentences is consistent," for example, defines the class of all formal systems in which the respective sentences are consistent. Inference rules in metamathematical systems, then, correspond at the semantic level to such things as transformations between classes of formal systems. Consider, for example, the compactness theorem "A set of sentences is consistent if each nonempty, finite subset of sentences is consistent." In this case, the class of formal systems defined by the premises can be transformed by logical procedures into the class of formal systems defined by the conclusion.

Statements in model theory denote relationships between formal systems and algebraic systems. Consider Gödel's completeness theorem "A theorem can be deduced if and only if it is a tautology." This theorem denotes the binary relation consisting of all pairs in which the first element is a statement (i.e., theorem) in the first order predicate logic that is deducible from the logical axioms; the second element is the corresponding property that holds in all systems. Since theorems are usually proved by arguing from premises to conclusions, inference rules in model theory may correspond to transformations between classes of formal systems and families of mathematical systems. In Gödel's theorem, one argues from the class of theorems deducible from the logical axioms of the predicate logic to the class of properties (tautologies) which hold in all mathematical systems.35

It would be an interesting exercise to identify and classify the semantic interpretations of the various inference rules involved in metamathematics and model theory. In some ways, this should be easier than with axiomatic theories because there are fewer and simpler inference rules to contend with.

35Strictly speaking, in Gödel's theorem one argues between a class of statements in a particular formal system and a class of universally true properties (tautologies).
Chapter 7

MECHANISMS OF LEARNING AND PERFORMANCE

MEMORY-FREE THEORIZING IN STRUCTURAL LEARNING

In Chapter 2, we attempted to clarify the nature of rules and rule-governed behavior. Then, in Chapters 4, 5, and 6, we showed how knowledge (competence) can be characterized in terms of rules, and applied the ideas to mathematics. It would appear that we have covered a lot of ground in these chapters and, in a sense, this is true. But, very little was said about structural learning or behavior per se.

In the next four chapters, we consider some of the theoretical questions which must inevitably be asked if we are ever to have an adequate theory of structural learning and performance. Chapter 7 is concerned primarily with the mechanisms of learning and performance, together with empirical work conducted under memory-free conditions. Chapter 8 deals similarly with the problem of motivation. The idealized theories of Chapters 7 and 8 are formalized and extended in Chapter 9. Chapter 10 shows how the idealized theory of structural learning can be enriched to include memory.

One of the major reasons why we have been relatively unsuccessful in devising adequate theories of complex learning and behavior, I believe, is because we have tended to tackle the problem as a whole. With a few exceptions (e.g., Miller and Chomsky, 1963), the possible value of ignoring the effects of memory in theorizing about structural learning and behavior has not been taken seriously by most psychologists. Oddly enough, however, this approach has a broad appeal to neophyte behavior theorists, particularly those with a mathematical or logical orientation. They have tended to attribute more in the way of rationality to human beings than the melange of available empirical evidence would seem to suggest.

Mindful of the early confusion between logic and psychology, most of us have tended to discourage this sort of activity, sometimes with almost
religious fervor. Although I know of no place where the reasons have been fully or clearly documented, many presumably feel that there is no theorizing to be done at the "idealized" level. The problems are viewed simply as ones of logical analysis. No theories or psychological assumptions need be made nor tested.

We reject this notion categorically. The present view is that some of the most interesting and basic psychological mechanisms are to be found at this level and, further, that these same mechanisms may carry over to the nonidealized situation as well. (See Chapter 10.)

Our approach to theory construction is clearly one of information processing. But it is more than that. We seek a theory about individuals, a theory which can predict individual behavior—and predict it in deterministic fashion. Although this goal seems almost unrealistic with respect to most traditional experimental situations, the evidence we have collected over the past few years strongly suggests that dealing with structured knowledge is intrinsically far simpler, say, than dealing with lists of nonsense syllables. In the latter case, the difficulties involved in identifying the critical individual differences appear to be almost insurmountable. The bases for associating two nonsense syllables, for example, may range over individuals and situations from keying on first letters to being reminded of (the subject's) "Aunt Tilly sitting on a pink Cadillac." Given such variation, it is hardly surprising that stochastic theorizing has been the rule. Asking a psychologist to make deterministic predictions about paired-associate learning would be much like asking a physicist to pinpoint precisely where particular leaves will fall on a windy day. In each case, known mechanisms may enter into the situation in so many different and unpredictable ways as to preclude definitive explanation or prediction.

In structural learning, it appears to be relatively much easier to identify and/or manipulate relevant knowledge (e.g., see Gagne, 1962; Scandura, 1968, 1969a, 1969b). (Also see Volume II, Chapter 2.) Idiosyncratic tendencies on the part of subjects seem to play a far less significant role.

This chapter is devoted to the following basic questions: (a) How can we find out what a given subject knows? and (b) Given a subject's goal and what he knows on entering into a situation, what are the mechanisms which govern learning and performance? More specifically, we shall consider: (1) What is the nature of decision making capabilities? Goals? Rules? (2) How are learned rules put to use? (3) How can one determine
what rules a subject knows (i.e., how to assess behavior potential)?

(4) What are the mechanisms by which new rules are acquired? Particular attention is given to the mechanisms of problem solving. (5) How can discovery learning be analyzed in terms of the postulated mechanisms? (6) Can expository learning also be analyzed in terms of these mechanisms?

1. SOME PRELIMINARIES—DECISION MAKING CAPABILITIES, GOALS, AND RULES

In view of their importance in the idealized theory, it may be helpful to say a few words about decision making capabilities, goals, and rules.

Decision making capabilities correspond to decision rules (tests) and refer to the ability to determine whether or not any one of a class of entities satisfies some (specified) condition. One such ability, for example, might involve distinguishing between those geometric figures in a set which are triangles and those which are not. Another might be concerned with whether or not given number series are arithmetic.

Although we did not stress the point in Chapter 2, decision making capabilities also serve to determine whether or not stimuli lie in the respective domains of rules. Number series, for example, are ordinarily put in the domain of rule

\[(A + L)/2\]

if and only if they are arithmetic. We note parenthetically that subjects frequently apply rules incorrectly. In general, this will happen whenever the initial decision making capability associated with a rule does not conform to the rule's true domain of applicability. In a study by Scandura, Woodward, and Lee (1967), for example, subjects frequently applied rule (1) to number series which were not arithmetic. Hence, the responses they generated in these cases did not satisfy the criteria used to evaluate them. (The responses were supposed to be sums). Subjects may also learn rules with overly restrictive domains, as shown in a study by Scandura and Durnin (1968). Summaries of both experiments are given in Volume II, Chapter 2.

In addition to serving as components of rules, decision making capabilities are needed to define goals, which we denote G. Indeed, a subject can hardly be said to have adopted a goal unless he has available a corresponding decision making capability. Unlike such capabilities, however, goals are imperatives to find (generate) responses which satisfy particular conditions. One goal, for example, involves finding sums of number series. Since any natural number may serve as a sum, of course, we ordinarily specify a particular stimulus situation in addition to the goal. The resulting pair is referred to as a goal situation and denoted \(\langle S, G \rangle\).
"Find the sum of 1 + 3 + 5 + ... + 99" provides an example. In order to satisfy this goal situation, the response must not just be a sum (natural number) but the sum of 1 + 3 + 5 + ... + 99.

To simplify the discussion, we have implicitly assumed that all decisions are of a binary (yes - no) nature. It is often necessary, however, to decide which of several possible conditions is satisfied. One might want to classify, for example, a set of geometric figures (responses) according to the number of sides.

It is well known that any multiple decision of this sort can be reduced to a sequence of binary decisions. One simply considers, in turn, each condition and its complement. Nonetheless, there are certain advantages in representing multiple decisions directly. For one thing, this makes it possible to represent effective procedures in *graph-theoretic* terms in a particularly useful way. (See Chapters 2 and particularly, 5.) The nodes of each characterizing graph correspond one-to-one to decision points in the corresponding procedure and the arcs all correspond to operations. If only binary choices are allowed, some arcs may simply indicate a movement from one decision point to another, with no intervening operation.

For experimental purposes, it seems advisable to distinguish among three types of goal condition. Two types are intrinsic to the task, and satisfaction of the condition, or a lack thereof, can be determined independently of external feedback. In one type, goal criteria are defined directly in terms of procedures which generate the responses. That is, a response satisfies the goal if and only if a particular procedure was used in generating that response. As an example, again consider the task of finding sums of long arithmetic number series. In this case, perhaps the simplest way to define the goal criterion is in terms of rule (1). That is, the goal is satisfied if and only if rule (1) was used to generate the response. (There are, of course, many alternative procedures for determining sums, but few are practicable with long series.)

In other situations, the goal conditions are defined independently of procedures which may be used to satisfy them. Consider, for example, the task of constructing a triangle, given the lengths of the three sides. One can determine whether a given triangle meets the desired condition by simply measuring the three sides, but this says nothing about how to construct the triangle in the first place.¹

¹A procedure which works is: Lay off one of the sides; call it A. Set a compass to radius equal to side B and draw an arc, using an end point of A as the center. Next, set the compass to radius equal to (continued)
A similar situation exists in proving theorems in mathematics. Here, it is one thing to determine whether a given proof is valid and quite another to construct one. Proofs can be checked by simple mechanical means, but as we saw in Chapter 6, they cannot be so easily or efficiently constructed.

In most structural learning, goal conditions that are intrinsic to the task itself appear to be the rule. The experimental literature on learning, however, deals primarily with experimenter-determined conditions, conditions which are usually defined in terms of extrinsic reinforcement (e.g., knowledge of results). Traditional concept attainment tasks (e.g., Bruner, Goodnow, and Austin, 1956), for example, are of this type; so are most of the tasks used in my own research on rule learning (cf. Volume II, Chapter 2). In these experiments, the subject has no way of checking himself; he is dependent entirely on what the experimenter tells him. The decision making capabilities involved in experiments of this sort, then, typically amount to nothing more than simply determining whether or not particular responses are compatible with what the experimenter says. Although the tasks used in such research have the disadvantage of being atypical (for one thing, they are usually artificial), they are, nonetheless, often easier to work with experimentally.

Decision making capabilities and goals may apply to rules as well as simple responses. In this case, the subject is required to judge whether or not given rules satisfy some specified higher order condition. In general, higher order conditions require that the domain of a rule contains some specified input, and that every element in its range satisfies some lower order condition. The task of finding a formula (rule) for summing a given arithmetic series provides a ready example. Finding a logical procedure by which some (logical) implication can be generated from a given set of premises provides another. It is important to emphasize that the desired goals in these examples involve formulas and logical procedures, respectively, and not sums or implications. As we shall see in Sections 4, 5, and 6, higher order goal criteria of this sort play a central role in problem solving and learning. The relevant ideas are formalized in Chapter 9.

So far in our discussions, we have not made an explicit distinction between the terms "rule" and "procedure." Nonetheless, we have tended to use the term "procedure" where we were concerned with internal structure

1(continued) side C and draw an arc, using the other end of A as the center. Finally, draw straight line segments from the end points of A to the intersection of the two arcs.
(i.e., with the constituent operations and decision making capabilities) as well as the procedure itself. The term "rule" has been used where the internal structure was not of concern.

In this chapter, we shall make another informal distinction. The term "rule" is used when the procedure in question has already been mastered by the subject. "Procedure" is used when we wish to speak more generally, as, for example, where we want a subject to learn a new procedure.²

2. PERFORMANCE MECHANISMS:
CONDITIONS UNDER WHICH (LEARNED) RULES ARE PUT TO USE

At any given point in time, human beings have a large number of learned rules available, but only a small fraction of them are applicable in any given situation. In this section, we ask: What are the necessary and sufficient conditions under which learned rules are put into use? Or, to put it differently, what is the nature of the performance mechanism?

Since we are concerned in this chapter with memory-free theorizing, we shall assume in asking this question that all of the potentially useful rules that have been previously learned are immediately available to the subject. That is, he does not have to retrieve them from memory; or, equivalently, there is no doubt that he can do so when required by the situation. (For a discussion of retrieval mechanisms see Chapter 10.)

Perhaps the simplest assumption one can make in this case is that human beings are goal-directed information processors. That is, the rules they select for use are determined by the goal situation at hand. This idea is obviously not new and goes back, even before Tolman, and more recently, Miller, Galanter, and Pribram (1960), to Gestalt psychology in the 1920's and William James in the 1890's.

Although the proposed performance mechanism has much in common with Miller et al.'s (1960) TOTE unit, however, it does differ in some important respects. The TOTE mechanism, recall, goes something as follows. First, the stimulus is tested against the goal criterion (supposedly to determine the size of the gap). Then, the stimulus is operated on (by a learned rule) to reduce the difference, and, finally, the output is again tested against

²It should be fairly evident that operations and decision making capabilities are closely related. For every operation, there is a corresponding decision making capability, and every decision making capability can be thought of as an operation. Although it is beyond the scope of this book to consider the question, there is some reason to believe that in human behavior decisions are affected (essentially) instantaneously, whereas operations take place in real time. Some further remarks to this effect are made in Chapter 9.
the goal criterion. The subject presumably continues this process until he either succeeds or runs out of rules to try.

In our case, it is assumed that for each subject there is a subset of learned rules attached to each goal condition. This subset may be thought of as consisting of all rules which may be used to satisfy the goal. Once a stimulus situation has been specified, one can reduce this subset of rules further to include only those rules whose domains contain the given stimulus. It is perhaps with this in mind that Miller et al. (1960) proposed an initial test. It is important to note, however, that this test is against the domain criteria of particular rules and not the goal condition.

The basic performance mechanism itself goes as follows: Assume that a subject is placed in a given goal situation \((S, G)\) (and has adopted the goal). Assume also that every rule known to the subject, which contains \(S\) in its domain and is such that each response in its range satisfies \(G\), is immediately available to the subject. Then, it is assumed that the subject will use one of the learned rules attached to the goal situation. The responses so generated are tested against the goal criterion as in the TOTE mechanism. If the goal is satisfied, then the subject exits to the next goal situation. If not, then the subject is assumed to try another learned rule associated with the goal situation until he either succeeds or runs out of rules.

Assuming that the subject has all the time he needs, the effect this mechanism will have on performance is summarized in the following simple assumption, which we state in preliminary form.

If a subject has adopted the goal \((S, G)\) and has a non-empty set of learned rules \(\{r_i \mid i \in I\}\) available, where \(S_o\) is in the domain of \(r_i\) and every response in the range of \(r_i\) satisfies \(G\) for all \(i\), then the subject will actually use one of these rules.

This may appear to be a trivial assumption about behavior, but it is nonetheless an assumption. It does not follow logically, just because a subject wants to achieve a certain goal, and has a class of learned rules available for achieving it, that he will necessarily use one of them.

A major purpose of the ensuing discussion is to show how the general point of view represented by the simple performance mechanism above may lead to: (1) useful predictions about complex human behavior, and (2) other assumptions which may play an even more crucial role in cognitive theorizing. It must be emphasized, however, that this assumption, together with those that follow in this chapter and the next, are reformulated
178 Structural Learning

slightly in a more general and rigorous way in Chapter 9. The treatment in Chapters 7 and 8 is retained for its heuristic value.

3. ASSESSING BEHAVIOR POTENTIAL

According to the behavior assumption of Section 2, if we know that a subject has a rule available for achieving a given goal, then we can predict his behavior perfectly. Ordinarily, however, we do not know this. Indeed, one of the main tasks to which our theory must address itself is that of how to determine what rules given subjects know.

Assessing what a subject knows is always relative to a given rule (procedure) set. Thus, given a potentially large and diverse class of behaviors (S-R pairs) of potential interest, and a finite procedure set which accounts for this class, one major task of the behavior theorist is to determine which parts of which procedures in the set each given subject knows. In accordance with our simple behavior assumption, this is equivalent to determining the subject's behavior potential (i.e., finding out which of the problems associated with each procedure in the given set the subject can solve and which problems he cannot solve).

The precise nature of the problem and its theoretical resolution is presented in Chapter 9, Section 2. In this section, we sketch the developments which eventually led to this formulation. First, we sketch some of the earlier research which led me to the question in the first place. This is followed by an operational definition of "what (rule) is learned," and a discussion of its relationship to assessing behavior potential.

Third, we show how the simple theoretical assumption of Section 2 provides an adequate basis for assessing behavior potential, or equivalently, what a subject has learned, given a knowledge only of the relevant prerequisite rules that the subject knew before learning. A substantial amount of supporting data is then presented. Finally, we summarize some recent data which shows further that given a characterizing set of procedures and the behavior assumption, it is possible both in principle and in practice to determine which behaviors, relative to each procedure, a given subject can generate and which he cannot. From this, and knowing the corresponding

Note that it is one thing to devise a procedure that accounts for a given class of rule-governed behaviors, and quite another to identify the subclass of these behaviors that a given subject can generate. The first problem is strictly analytical in nature and involves inventing a procedure(s) which accounts for the given class of rule-governed behaviors. No psychological assumptions are involved. Finding a solution to the latter problem, however, will necessarily depend on what can be assumed about the mechanisms which govern human behavior.
procedures in the characterizing set, it is possible to construct a rule which accurately represents what the subject knows.

3.1 Related Research

I first became concerned with the problem of assessing behavior potential during the summer of 1962, when Greeno and I (1966) found in an experiment on verbal concept learning that subjects either give the correct response the first time they see a transfer stimulus or the transfer stimulus is learned as its control. The thought later occurred to me (Scandura, 1966) that if transfer obtains on the first trial, if at all, then responses to additional transfer items, at least under certain conditions, should be contingent on the response given to the first transfer stimulus. In effect a first transfer stimulus could serve as a test to determine what had been learned during the original training. It would then be possible to predict a subject's response to a second transfer stimulus.

Since that time, some of my students and I have collected a fairly substantial body of data which provides strong support for this contention (Scandura, 1966, 1967b, 1969a; Scandura, Woodward and Lee, 1967; Scandura and Durnin, 1968; Roughead and Scandura, 1968). In these studies, the subjects were presented with a rule statement or a number of instances of a rule and required to learn the underlying rule. The subjects were then presented with a test stimulus and instructed to respond on the basis of what they had just learned. They were told they were correct no matter what the response. Then, they were presented with a second test stimulus.

The results have generally been quite clear-cut. Whenever the response given by a subject to the first test stimulus was in accord with a particular given or derived rule, so was the response to the second test stimulus. We generally have been able to predict second test behavior with anywhere between 80 and 95 percent accuracy.

The results of one study (Scandura and Durnin, 1968) on extra-scope transfer further suggest that individual subject's behavior potential can often be determined by the appropriate selection of just two test instances. In particular, Scandura and Durnin found that successful performance with two stimuli which differ along one or more familiar dimensions implies successful performance with other stimuli which differ only along these dimensions. This result suggests that success on well-chosen test instances may be adequate to determine which S-R instances in a given class a subject is potentially capable of generating and which he is not. The question remains, of course, as to what is meant by a well-learned dimension (or,
well-chosen test instance).

Although our work progressed independently, and with somewhat different objectives, it was encouraging to note that Levine (1966) and his collaborators (Levine, Leitenberg and Richter, 1964) also found this consistency notion useful. Levine, Leitenberg and Richter, for example, used performance on nonreinforced trials to predict performance on reinforced trials with a high degree of success.

In general, it would appear that when a subject thinks he is right and the new situation remains relevant, he will continue to respond in a consistent manner.

I originally thought that the entire notion of response consistency involved capitalizing on Einstellung (a mental set to perform) to ascertain what is learned and to predict performance on future items. As it turns out, this is incorrect. The hypothesis is too restrictive. It does not appear necessary to assume that a subject will continue to use the same rule on all test instances. Rather, we can simply assume, in a given situation, that if a subject has one or more appropriate rules available, then he will use one of them. This allows a subject to use one rule on one test instance and another one on a different instance. For example, in adding numbers, a subject might apply the ordinary addition algorithm to a problem like $726 + 398$, whereas a doubling technique of sorts might be used with a problem like $250 + 250$.

3.2 The Problem

With this in mind, let us turn more specifically to the problem at hand: how to determine a subject's behavior potential, relative to a given class of rule-governed behaviors, on the basis of the subject's performance on a (small) finite number of test instances. To make the discussion definite, we may think of a particular class of S-R pairs or instances—where the stimuli, say, are arithmetic number series and where the responses are the corresponding sums.

The definition of "what rule is learned" in Scandura (1970b), takes a step in the right direction. The basic idea is that given a performance profile, characterized by success on $m$ of $n$ instances in a given class and failure on the remaining $n-m$ instances, then "what is learned" may be defined as the class of rules which provides an adequate account of the data.

To see how this definition applies, consider the class consisting of arithmetic number series and their respective sums. First suppose that a subject has demonstrated his ability to find the sum (2500) of the arithmetic
series $1 + 3 + \ldots + 99$. The definition tells us that the class "what is learned" includes all and only those rules which provide an adequate account of this behavior. In this case, the class would include, among possibly other rules, each of the following: sequential addition (applied to arithmetic number series); the general rule $\left(\frac{A + L}{2}\right)N$ for summing arithmetic series; the rule $N^2$, which applies to all arithmetic series of the form $1 + 3 + \ldots + (2N-1)$; the direct "association" between the series $1 + 3 + \ldots + 99$ and its sum 2500. Thus, "what is learned" might be denoted by the class

$$\{\text{direct association, } N^2, \left(\frac{A + L}{2}\right)N, \text{sequential addition, } \ldots\}$$

As more test information is obtained about a subject's performance capability, it will generally be possible to eliminate rules from this class. Suppose, for example, that a subject is successful in determining the sum not only of the original test series but also, say, of the series $1 + 3 + \ldots + 47$. Then the size of the class "what is learned" is reduced accordingly to

$$\{N^2, \left(\frac{A + L}{2}\right)N, \text{sequential addition, } \ldots\}$$

The direct association would no longer be allowed because it does not apply to the second series. If the subject is successful on still another test instance, say, on the series $2 + 4 + \ldots + 100$, then the class "what is learned" is further reduced to the set

$$\{\left(\frac{A + L}{2}\right)N, \text{sequential addition, } \ldots\}$$

Suppose, on the other hand, that the subject is successful on the first two test stimuli (i.e., $1 + 3 + \ldots + 99$ and $1 + 3 + \ldots + 47$), but not the third (i.e., $2 + 4 + \ldots + 100$). Then, according to the definition, not only would the direct association be eliminated as a feasible rule, but so would the more general rules $\left(\frac{A + L}{2}\right)N$ and sequential addition. In effect, the class "what is learned" would include only $N^2$, together with possible other unidentified rules which provide an adequate account of the behavior.

As indicated, this definition provides a way for deciding whether or not a rule is to be included in a class. It can even be useful in making predictions about behavior potential. The definition, however, says nothing about how rules in the class are identified in the first place. More important, the definition provides no basis for identifying the test instances needed to determine behavior potential. To specify the behavior potential of an individual relative to a given class of behaviors, one must be able to specify a finite subset of test instances which make it possible to predict the subject's performance on all of the remaining items.
The *traditional* approach to this problem has been to resort to probability statements of one sort or another. Thus, predictions about behavior are made on the basis of past performance on a random sampling of test items. In this case we refer to such statistics as expected values and variances in making predictions.

The point of view taken here is that under memory-free conditions, there is no need to resort to probability in describing the behavior potential of human beings. This does not imply perfect prediction, but only that uncertainty, rather than being an explicit part of the theory or prediction model, has its source elsewhere. Perfect prediction is impossible in principle because there is always some residual uncertainty about the capabilities and motivations that a subject brings to any given task. And, we feel that this is exactly where this uncertainty should be put.

Rather than make probabilistic predictions, based on a random sampling of test instances, we propose an approach to assessing behavior potential that makes deterministic predictions possible. The basic idea involved in selecting an adequate set of test instances is to *partition* the given class of behaviors (S-R pairs) into a set of mutually exclusive and exhaustive subsets so that each subset of instances is atomic. By an atomic subset is meant an equivalence class in which success on any one instance in the class is indicative of success on any other instance in the class, and similarly for failure. Studies conducted by Scandura and his students and others (e.g., Restle and Brown, 1970) over the past several years strongly suggest the existence of such equivalence classes. Scandura and Durnin's (1968) subjects, for example, almost always performed uniformly (well or poorly) on classes of similar items.

Once such a partition has been found, of course, the next step is obvious. One simply selects one test instance for each element (i.e., equivalence class) in the partition. Predictions on new instances, then, are made in accordance with whether or not the subject succeeds on the corresponding test instance—that is, the test instance in the same equivalence class of the partition.

Unfortunately, the problem of how to determine such a partition in the first place is not trivial. Some idea of the difficulty may be seen by considering simple addition. Any attempt to identify equivalence classes of addition problems, immediately raises such questions as whether $25 + 30$ is more like $5 + 30$ or more like $20 + 30$. Clearly, some alternative to sheer guesswork is needed.
3.3 A Solution

After mulling this problem over for some time, I concluded that the most feasible way of identifying such a partition is to tackle the problem intentionally—in terms of underlying procedures. (This does not rule out the possibility that an extensional or dimensional analysis may also work.4) The problem here is one of reducing the number of possible procedures to reasonable proportions. Theoretically, there are an infinite number of procedures which might account for any given class of rule-governed behaviors.

Fortunately, this does not seem to pose undue difficulty in practice. Given any familiar rule-governed class, it is usually possible to determine those procedures which the subjects in question are most likely to use. There are relatively few basically different procedures for adding fractions, for example. Furthermore, given any algorithm it is always possible to break the procedure down into what might be called atomic (sub)rules. By an atomic rule is meant a rule which, if it elicits success on one instance, implies success with the others, and similarly for failure.

To see this, it is sufficient to recall from Chapter 2 that, given any rule, it is always possible to break it down far enough so that the constituent rules act in atomic fashion. This is trivially true in the case where the constituent rules are associations. (For a more formal treatment consult Chapters 5 and 9.)

Furthermore, given even the barest minimum of information about a subject's capabilities, it is usually possible in practice to make intelligent guesses concerning whether or not a given subrule is apt to act in atomic fashion. Thus, for example, in subtracting whole numbers, most subjects either know how to borrow, or they do not. There is no in-between.

The precise nature of the relationship between a procedure and the atomic rules of which it is composed can perhaps be seen most easily by representing the former as directed graphs in which arcs correspond to atomic rules in the procedure, and nodes to branches. For example, consider the procedure for generating the "next" numeral in base three. The atomic rules are as follows.

4Originally, I proposed that behavior potential might be defined relative to certain (test) stimulus dimensions which were assumed to be well-defined for the subject (cf. Scandura and Durnin, 1968). Although developed on more pragmatic grounds, the item form method proposed by Hively, Patterson, and Page (1968) also addresses itself to this problem. It now appears to me that both types of extensional analysis are more naturally interpreted in terms of underlying procedures.
(1) Read (encode) the one's digit of the given numeral.
(2) If the digit is a "2," change the "2" to "0," write it down (and go to 4).
(3) Increment the digit by 1, write the new numeral (and stop).
(4) If there is another digit to the left, encode it (and go to rule 2).
(5) Write "1" in the next position to the left of the last "0" written (and stop).^5

The S-R instances generated by this procedure would be of the following sort: 0→1, 1→2, 2→10, 10→11, ..., 1022→1100, .... The procedure may be represented by the directed graph

where the individual arcs of the graph correspond to atomic rules (which correspond to steps in the algorithm). In this particular case there are four distinct paths through the algorithm

When looked at in this way it is easy to see that a person might master certain paths of a given procedure but not others. Furthermore, given that the rules in such a procedure (including the decision rules) are atomic, one can logically conclude that each path in the procedure must also act in atomic fashion. From this it follows that each procedure effectively partitions the original rule-governed class into equivalence classes of the type mentioned earlier. In effect, an observer can determine which paths a subject has learned, and which paths the subject has not, by observing the subject's performance on a single instance from each equivalence class in the partition.

Of course, more than one procedure may underlie the same rule-governed class. Essentially the same approach applies here, however, because the

---

^5Strictly speaking, steps 2 and 4 in this procedure combine operations with decision making capabilities.

^6Durnin (personal communication) has found that the number of different paths through a procedure and, hence, the number of elements in the partition, depends on the number of branches in a very direct way. If a procedure has N binary branches, then it must have at least N + 1 but no more than $2^{2^N}$ different paths.
intersection of the partitions that are associated with the various procedures is also a partition. In this case, it is the intersection partition we use to select the test instances--one test instance for each element in the intersection partition. In the "next numeral" example above, the numerals 101, 2, 112, and 222 provide an adequate set of test stimuli. Each is associated with a distinct equivalence class. (In this case, of course, the partition is determined by a single procedure.)

The hypothesis which underlies assessing behavior potential, then, may be stated as follows: Given the simple performance hypothesis and assuming that the constituent rules of the procedures associated with the rule-governed class all act in atomic fashion, then if a subject can solve any task in a given equivalence class of the intersection partition, he should also be able to solve any other task in the same equivalence class--and similarly for failure.

This intentional approach also provides a systematic way of devising a set of procedures which directly parallels the subject's behavior potential --that is, for constructing procedures which generate exactly those S-R instances on which success can be expected. This can be accomplished by deleting those atomic rules of the original procedures which contribute only to the generation of incorrect responses during testing.

Implicit in this process is the assumption that the paths of a given procedure can be partially ordered. A path which contains all of the atomic rules of another path plus some of its own would occupy, relatively speaking, a higher position in the ordering, because if the latter path contains a deleted atomic rule, then the former path surely will. From this it follows (on largely logical grounds) that if a subject is successful on an instance associated with a higher level path, then the subject should also be successful on items associated with all (relatively) lower level paths. This provides a second hypothesis that can be tested experimentally.

More precisely, "intersection partition" refers to that partition which consists of all n-fold intersections of equivalence classes involving one equivalence class from each of the n partitions associated with the n nonequivalent rules. For details, see Chapter 9.

The usefulness of this approach in future research, as well as practice, requires more than just testing the theoretical hypothesis upon which it is based. With experimental confirmation of the simple performance hypothesis in the assessment situation, research can proceed in several directions. First of all, it would be important to determine the value of this procedure in assessing the behavior potential of individuals at different developmental levels and with a variety of different kinds of meaningful tasks. This could be particularly important for research in various subject matter areas such as mathematics. Second, it might be used to determine individual differences in computational ability so that this might be taken into account in studies of problem solving in (continued)
3.4 Testing the Hypothesis

The basic (first) hypothesis above rests on two major premises: (1) The simple performance hypothesis of Section 2 holds. (2) All of the procedures that might possibly be used by a subject have been identified, and represented so that each component subrule acts in an atomic fashion. Given these premises it is possible both to predict the subject's behavior potential relative to the given rule-governed class, and to determine which of the identified procedures, or parts thereof, that individual subjects know.

In testing this hypothesis, we implicitly assume that the simple performance assumption holds—without formal empirical testing. That is, we assume that if we somehow knew ahead of time what relevant rules a subject had available, then we could predict his behavior perfectly. Our concern is almost exclusively with the second premise, where we have less than complete information about what given subjects actually know.

A direct test of the second premise would involve situations where we know explicitly what procedures may be used and that these procedures have been represented in terms of atomic rules. Although this ideal can only be approximated in practice, it can be approached with novel tasks. Here, the task solutions are defined in terms of arbitrarily constructed procedures that the subjects could hardly be expected to know ahead of time. Subjects, then, can either be trained (to a high criterion), or not be trained, on individual components of these procedures so that each component subsequently acts in atomic fashion. Next, we can provide the subject with an opportunity to learn, according to as yet unspecified mechanisms, how to solve the tasks. After some degree of learning has taken place—i.e., after certain of the known atomic rules have been appropriately combined to form paths of a solution procedure, the hypotheses can appropriately be tested.

A series of mini-experiments is reported in the next section which demonstrate, fairly conclusively I think, the feasibility of the two basic hypotheses. Given all of the solution procedures likely to be used, and after training the subjects in the use of the atomic rules to a high

8(continued) which subjects have to compute. If the procedure can also be made to work with such tasks as number conservation and the like, developmental psychologists could have a valuable new tool for their research.

Insofar as practice is concerned, there are also some fairly immediate implications of such a procedure for diagnostic testing, sequential testing and computer assisted instruction. Research in some of these areas will be described in future reports.
criterion, we provided the subjects with an opportunity to learn as much as they reasonably could be expected to learn with respect to the given task. A special (practical) problem in this regard was to devise tasks and procedures which were at an intermediate level of difficulty. If the procedures were too simple, the subjects learned all of the paths and if too complex, they learned none. Then the subjects were tested under non-reinforcement conditions on two items representative of each equivalence class in the associated partition. Performance on the first test items in the equivalence classes were used to pinpoint what had been learned, and the second items to determine the adequacy of the assessment. These mini-experiments were conducted with a variety of tasks under different conditions and with different age populations.

To illustrate, again consider the procedure for generating the "next numeral" in base three. (We assume the task to be unfamiliar to the subject(s) in question.) After identifying the paths of the procedure, and the four corresponding equivalence classes, two instances are arbitrarily selected from each. The numerals 101, 2, 112, and 222 are illustrative of the four equivalence classes.

In an actual experiment each subject would be trained on those atomic rules that we want the subject to enter the testing situation with. Thus, in the "next numeral" example we might have the subject learn atomic rules 1, 2, 3, and 4 but not, say, 5. The criteria used to insure that the intended learning of the atomic rules has indeed taken place would necessarily be to some extent arbitrary but this should pose little problem in practice. The main thing to insure is that the subject can perform perfectly on any instance of each atomic rule we wish to present him with. Consider, for example, atomic rule 1. Only the most determined nonbeliever would be unwilling to admit that a subject had mastered rule 1 if, after training, he is able to "read" correctly the last digit of a wide variety of numerals like: 2, 10, 110, 1010101, 112, 1111, 1012, 22222, etc.

Because the procedure is assumed to be unknown initially, the subject would then be provided with an opportunity to learn (parts of) the next numeral procedure. Although the mechanisms by which such learning takes place are not explained until Section 4, the results of previous research provide a sufficient basis for defining appropriate learning and testing environments. Specifically, it has been shown (e.g., Levine, 1966) that learning only takes place on reinforced trials, where the subject is told or can otherwise determine whether or not his responses are correct. After the subject is provided with an opportunity to learn, he is tested under
Experiments of this sort involve manipulating certain aspects of prior learning of individual subjects, providing an opportunity for learning, testing each subject, and then, making predictions as to what they will do on individual items. It is not a question of running groups of subjects or of averaging over different tasks. We experiment with a number of different subjects on a number of different tasks under a number of different conditions, to see if we can predict exactly what the subject can and cannot do.

Our predictions are tested by seeing in how many cases they hold and in how many cases they do not. Questions of statistical inference do not enter, at least not in the usual sense. Of course, we cannot expect to be right all the time, but we do want to come as close to that ideal as possible. Our earlier research on response consistency (referred to in Chapter 2, Volume II) suggests that four correct predictions out of five would be a good standard to apply.

In view of the deterministic goal of this research, however, it would not be our intention to simply attribute the "fifth" case to random error, but, where possible, to explain why the results did not turn out as expected. In this sense, our goals are much like the early physicist's. Where a law failed, he tended to attribute the result (where possible) to inadequate controls of one sort or another. His reasons typically ranged from unwanted friction or air resistance to an unknown additional force or forces acting on the body in question. In our case, the corresponding reasons might range from inadequate control over the environment to the influence of unidentified learning. In reporting actual results, of course, it might be useful to compute confidence intervals concerning the percentage of "hits" and "misses." In combining this paradigm with that introduced and well-documented by Levine (e.g., 1966), it is possible to study the actual course of learning. Although Levine's research has dealt exclusively with simple discrimination learning, the basic ideas can apparently be extended to more complex situations as well. Stated in terms relevant to our purposes, the

9These observations are but special cases of a discovery by philosophers of science (Ossorio, unpublished communication) that scientific laws cannot be demonstrated to be false. There is always an out. The point of experimentation is to determine how useful a scientific law is—that is, how widely applicable the law and how succinctly it summarizes the data to be explained.

10An interesting new approach to hypothesis testing that seems to be compatible with present requirements can be found in Hildebrand, Laing, and Rosenthal (1971).
main ideas are as follows. (1) After pretraining on the atomic rules (hypotheses), the subject is provided with an opportunity to learn. That is, the subject is presented with test instances of the given rule-governed class and given feedback (i.e., told whether he is right or wrong). 11 (2) Periodically during the course of learning, the subject is tested without reinforcement by a procedure that is directly analogous to the assessment procedure described above. (According to our own research findings and those of Levine's, a subject will not learn under the proposed conditions of nonreinforcement.) Unlike Levine's procedure, however, ours is perfectly general. It should also be noted that whereas we introduced a second set of test items to determine the adequacy of the original assessment, Levine did not.

Although it is possible to gain information about the course of learning in this way, such information is necessarily probabilistic. Changes in learning from stage to stage, as in the research of Levine (1966) and Restle (1962), for example, have been essentially stochastic phenomena. (In Section 4, we shall be concerned with specifying precisely what a subject needs in order to learn at each stage.)

One further point is worth mentioning. Although the research of Levine, Restle, and others of that persuasion has included important elements of the type of analytical method we are proposing, they have not attempted to eliminate memory load. To the contrary, their theory (Hypothesis Theory, as it has been called) has been designed to take the subject's limited capacity to process information into account. This confounding of theorizing at the memory-free and memory-dependent levels has, in my opinion, led learning theorists to an almost complete reliance on stochastic theories.

It is one thing, of course, to test the assessment hypothesis directly, and quite another to determine what a subject knows on entering a situation. In the latter case, assumptions must be made not only about what procedures are viable, but also about the level of refinement required so that the components of these procedures act in atomic fashion. 12

Where the tasks and procedures involved are familiar, what makes this feasible is the common culture shared by the experimenter and subject alike.

---

11 There is, however, a major difference between the type of "learning" involved in the Levine studies and that proposed here (cf. Chapter 8).

12 Assuming that all of the viable underlying procedures have been identified, the predictions should be adequate to the extent that the presumed atomic rules are indeed atomic. One would not expect good predictions and constituent rules which are not atomic, or poor predictions and constituent rules which are atomic.
To the extent that their backgrounds differ, the more difficulty one might reasonably expect in identifying viable procedures, and in representing them appropriately. As Piaget's research so clearly shows, for example, the rules which govern the behavior of young children are sometimes quite foreign to adults.

Durnin and Scandura (1971) recently completed a study which not only demonstrated the feasibility of this algorithmic approach to assessing behavior potential, but also compared it with two other approaches—item forms and hierarchical analysis—that have recently been proposed on more strictly pragmatic grounds. The basic results as they involved the algorithmic approach are summarized in Section 3.6 Assessing Behavior Potential.

3.5 Some Empirical Results

Scandura and Durnin (1971) have completed a number of experiments designed to test the two basic hypotheses of Section 3.3 (see p. 185). The approach was as follows.

1. A suitable class of rule-governed behaviors (i.e., S-R pairs) was selected, together with an algorithm for generating each behavior in this class. That is, given any stimulus in the class of stimuli, the corresponding response could be generated by following a path of this algorithm. It was of particular interest to choose tasks which were not so easy that the subjects were uniformly successful on the task, nor so hard that they were uniformly unsuccessful.

2. Each of the atomic rules included in this algorithm was then built directly into the subject in question in the sense that he could correctly apply each of the constituent atomic rules uniformly well. Where the number of instances of an atomic rule was infinite, enough varied instances of the rule were chosen to assure ourselves that the subject did have complete mastery.

3. Armed with these atomic rules as prerequisite knowledge, the subject was presented with stimulus instances of the given rule-governed class and was asked to generate the corresponding responses. After the subject gave his response, or indicated that he did not know the answer, knowledge of results was given in either one of two forms—self-reinforcement or external reinforcement provided by the experimenter. This procedure was continued with new instances until the subject seemed not to be making any further progress on the learning trials. During the learning period, care was taken to insure that the subject had ample opportunity with instances corresponding to each path of the algorithm.
4. The subject was then tested on one arbitrarily selected instance for each path. More accurately, the instances were selected from the equivalence classes corresponding to the various paths through the algorithm.

5. Predictions were made about the subject's performance on future instances associated with these equivalence classes (paths of the algorithm) on the basis of how the subject had performed on the test instances. These predictions were checked in the obvious way by simply testing the subject on another set of representative instances.

According to our analyses, confirmation was expected with all subjects on all tasks to the extent that the presumed boundary conditions were met—that is, to the extent that the subject was actually trying to solve the problems, the presumed atomic rules were indeed atomic, and the subject was unencumbered by memory or his limited capacity to process information. Our aim was to run this basic experiment with a wide variety of different tasks and with a wide variety of different subjects at different developmental levels. In particular, subjects were sampled from graduate school down to the preschool level. Notice that each combination of task and subject constitutes a replication of the basic experiment.

The task and method used in experiments 1 - 5 are described in detail. The others are only summarized, except where we wish to emphasize a particular point. (For details, see Scandura and Durnin, 1971.)

The task used involved self-reinforcement. That is, the subjects were able to check their answers independently during the learning period (but not during testing).

1. The class of rule-governed behaviors (task) used in these experiments involved finding multiplicative inverses of four-tuples (e.g., \((0, 1, 2, 0)\)) of integers modulo 3, and was modelled on matrix multiplication. The system of integers mod 3, which consists of elements 0, 1, and 2, may be defined by the addition and multiplication tables

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

The subject was shown how to multiply two four-tuples by using the definition \((a, b, c, d) \times (e, f, g, h) = (a \cdot e + b \cdot g, a \cdot f + b \cdot h, c \cdot e + d \cdot g, c \cdot f + \)

13This document will be available from the U.S.O.E. ERIC retrieval systems beginning September, 1973. Prior to this date please contact the author.
d-h) where + and \cdot, respectively, are the operations of addition and multiplication in mod 3. For example, \((1, 0, 2, 1) \times (2, 0, 1, 2) = (1 \cdot 2 + 0 \cdot 1, 1 \cdot 0 + 0 \cdot 2, 2 \cdot 2 + 1 \cdot 1, 2 \cdot 0 + 1 \cdot 2) = (2, 0, 2, 2)\).

A four-tuple \(A\) is said to have a \textit{multiplicative inverse} if there exists a second four-tuple \(B\) (possibly the same) such that their product is \((1, 0, 0, 1)\). For example, since \(A \times B = (1, 0, 2, 1) \times (1, 0, 1, 1) = (1, 0, 0, 1)\), \(B\) is said to be the multiplicative inverse of \(A\).

The subject's goal in experiments 1 - 5 was to find inverses (responses) of certain given stimulus four-tuples. If a four-tuple \((a, b, c, d)\) has an inverse, then the following flow diagram presents one algorithmic procedure for finding it.

![Flow Diagram](image)

This flow diagram can be represented more simply as a directed graph where lines represent operations (atomic rules) and points, the decisions which need to be made in carrying out the algorithm. This graph, together with the six possible paths through it, and illustrative stimuli associated with these paths, is given in Figure 2.

2. During the preliminary training phase, the subjects were first taught how to add and multiply integers mod 3 and an operation \(*\) defined by \(*(0) = 0, * (2) = 1\) and \(* (1) = 2\). They were also given the definition for multiplying four-tuples and practice in applying the definition. The
Subjects were also shown the identity \((1, 0, 0, 1)\) and told what an inverse was. This information made it possible for the subject to check his answers.

Then, each subject was taught the following six atomic rules.

1. Form the products \(a \cdot d\) and \(b \cdot c\) (i.e., multiply \(a, d\) and \(b, c\)).

   Example: Given \((1, 0, 2, 1)\) then \(a \cdot d = 1,\ b \cdot c = 0\).

2. Form \(a \cdot d + *(b \cdot c)\).

   Example: Given \(b \cdot c = 2,\ a \cdot d = 1\), then \(a \cdot d + *(b \cdot c) = 2\).

---

The numbers associated with the various arrows specify which atomic rules they represent.

The diagram indicates that there are several other paths through the procedure (e.g., \(\rightarrow\)). No instances in the given rule-governed class require the use of these vacuous paths, however; so they may be discarded.
(3) (a) Interchange a and d; i.e., (a, b, c, d) \rightarrow (d, b, c, a)
(b) Multiply d, b, c, and a by 2; i.e., (2d, 2b, 2c, 2a)
(c) Change 2b to \textit{*}(2b) and 2c to \textit{*}(2c)
   \textit{Example: } (2, 2, 1, 0)\text{ (a)} (0, 2, 1, 2)\text{ (b)} (0, 1, 2, 1)\text{ (c)}
   (0, 2, 1, 1)

(4) (a) Multiply a, b, c, and d by 2; i.e., (2a, 2b, 2c, 2d)
(b) Change 2b to \textit{*}(2b) and 2c to \textit{*}(2c)
   \textit{Example: } (2, 1, 0, 1)\text{ (a)} (1, 2, 0, 2)\text{ (b)} (1, 1, 0, 2)

(5) (a) Interchange a and d; i.e., (a, b, c, d) \rightarrow (d, b, c, a)
(b) Change b to \textit{*}(b) and c to \textit{*}(c)
   \textit{Example: } (2, 1, 0, 1)\text{ (a)} (1, 1, 0, 2)\text{ (b)} (1, 2, 0, 2)

(6) Change b to \textit{*}(b) and c to \textit{*}(c)
   \textit{Example: } (1, 1, 0, 1) \rightarrow (1, 2, 0, 1)

The subject was trained on each rule to a criterion of at least three correct responses in a row and was allowed as much time in working with the rules as he required. The order of presenting these rules was randomized.

3. After the subject had learned the six atomic rules to criterion, he was given a practice sheet consisting of 24 stimulus instances from the rule-governed class. The problems presented were divided into four sets of six instances each, one instance from each of the six equivalence classes. Within each set, the instances were randomized. The subject was then told to find the inverse of each four-tuple, using the (atomic) rules he had just learned. He was told to do as many of the problems as he could and to check his answers by multiplying the four-tuple he derived with the one given. In order to be correct, the product had to be (1, 0, 0, 1).

A printed statement of the rule for multiplying four-tuples was available to him at all times so that he did not need to commit the rule to memory. The subject was allowed as much time as he needed to complete the problems.

4. After the subject had completed as many problems as he could, the experimenter collected the problem sets and the printed statement of the rule for multiplying four-tuples.

The first set of test problems consisted of six new stimulus instances, one instance from each equivalence class. During the testing, as well as the pretraining, the subject had statements of the atomic rules available in case he forgot any of them. He did not, however, have the rule for multiplying four-tuples so that he could not check his answers. Indeed, the subject was closely monitored so that he did not have time to check
his answers even if he had succeeded in memorizing the rule. According to our earlier results, then, the subject was unable to learn during testing; the test trials correspond to Levine's (1966) nonreinforcement trials.

5. The subject was given a second set of test problems of the same type immediately following the first test. There were no time limits on either test.

The results of experiments 1 - 5 (five subjects using this task) are given below.

<table>
<thead>
<tr>
<th></th>
<th>1st test</th>
<th>2nd test</th>
<th>1st test</th>
<th>2nd test</th>
<th>1st test</th>
<th>2nd test</th>
<th>1st test</th>
<th>2nd test</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The "+" indicates a correct response while "-" indicates an incorrect response. The encircled pair indicates a result which went contrary to prediction. In this case, there were 29 correct predictions and one incorrect.

Experiments 6-21

The second set of experiments involved tasks in which the reinforcement was external. Only on the practice trials, did the experimenter tell the subject whether or not he was correct.

The task used in experiments 6-9 was adapted from Polish notation used in logic to avoid the use of parentheses (cf. McCall, 1967).

For example, subjects were given strings of symbols, like MA5,6 A2,1, and asked to compute the result given that A meant Add and M meant Multiply. In accordance with the algorithm identified, the first step in this case would be to add 5 and 6, then add 2 and 1, and finally multiply the two sums 11 and 3, giving the result 33. Eight equivalence classes of stimuli
were determined from the algorithm.

The subjects were 2 graduate and 2 undergraduate students at the University of Pennsylvania and the experimental procedure was identical to that described above.

The results were as predicted in 31 of 32 cases.

In experiments 10 and 11, the task was based on another method of multiplication (cf. Eves, 1964). To illustrate the method, consider the task of finding the product 456 x 32. First, construct the array

```
  4  5  6
 1  2
1  5
0  1
```
and record the individual products in the boxes as shown. The product 14,592 is determined by summing (within the large rectangle) along the diagonals.

```
  4  5  6
1  2  1  8
0  8  1  2
```

The specific procedure employed had eight paths.

Again the same experimental procedures were used, this time with two sixth-grade students in West Philadelphia.

Fifteen of 16 predictions were correct.

The task for experiments 12-15 was designed for use with preschoolers, and involved crossing out squares, circles, and combinations thereof from given displays of three circles and three squares (e.g., \( \text{\textcolor{red}{O}} \text{\textcolor{blue}{O}} \text{\textcolor{yellow}{O}} \text{\textcolor{red}{O}} \text{\textcolor{blue}{O}} \text{\textcolor{yellow}{O}} \)). The subject was given a stack of 3" x 5" stimulus cards with a colored figure drawn on one side of each card. The figures varied along two dimensions: color (red, blue and yellow) and shape (square and circle), with the relevant dimension being shape. The cards were turned face down. The subject was told to turn each card over and cross out on the display sheet whatever figure was shown on the card. The corresponding procedure had 5 paths.

Four children, aged 5, 4, 3, and 2 1/2, served as subjects.

The results were as predicted in all 20 cases.

During the course of experiments 12 and 13 we observed that the two older subjects who could count used a slightly different procedure than
the one we identified. Fortunately, the paths through the procedure they used partitioned the class of tasks into exactly the same equivalence classes. The hierarchical arrangement of the paths associated with these two procedures was different, however, and the effect of this on the hierarchy results is reported below.

The task used in experiments 16-19 was also designed for use with young children. In addition to crossing out squares and circles, as before, whenever two circles or three squares, respectively, were crossed out the subject was required to also draw a circle within a square or a square within a circle. Given the stimulus \( \bigcirc \Box \bigcirc \Box \bigcirc \Box \), for example, a correct response would be \( \bigcirc \Box \bigcirc \Box \bigcirc \Box \bigcirc \).

One 7-year-old, one 6-year-old, and two 4-year-olds served as subjects. Four equivalence classes of test instances were identified; 14 of the 16 predictions were correct. The two inconsistencies were attributed to the 6-year-old for whom single and repeated application of an atomic rule seemed to pose a different problem. Although the data do not speak to this point it seems reasonable to suspect that the decision making capabilities employed by this particular child were not adequate.

In experiments 20-21, the subjects were shown two cards containing one, two, or three congruent figures (circles, triangles, or squares) and were asked to identify which properties (number, shape) were common to both. This was indicated by turning over response cards indicating specific properties. The procedure for accomplishing this involved two atomic rules and four paths.

One subject was 5 years old and the other was 6.

Eight correct predictions were made out of 8.

Experiments 22-30

In the third set of experiments the subjects had no obvious way of knowing during practice whether their answers were correct or not. They were neither told when they were correct nor given an independent means for checking their answers. The main purpose was to determine whether the intrinsic structure of procedures may itself provide sufficient reinforcement for learning.

The task used in experiments 22-26 was based on forming products of 2-cycle permutations (cf. Herstein, 1964). The procedure employed had eight paths.

The subjects were four graduate and one undergraduate students at the University of Pennsylvania.
Correct predictions were made in 39 of 40 cases.

The task used in experiment 27 was essentially a complication of that used in experiments 16-19. The subject was an undergraduate at Pennsylvania.

The results were 7 predictions out of 8.

The final task was adapted from an old method for adding Roman Numerals (cf. Eves, 1964). There were 8 paths through the associated algorithm.

The subjects were two high school students in West Philadelphia and one undergraduate at Pennsylvania.

Correct predictions were made in 24 of 24 cases.

Discussion. The studies reported in the previous section provide strong support for the basic hypothesis of Section 3.3. In a total of 194 cases, there were 187 or 96% correct predictions. The 95% confidence interval for this proportion (96%) is between 93% and 99%.

The results also provide evidence in favor of an even stronger hypothesis. Not only does success on one instance of a given equivalence class imply success on any other instance in that class, but frequently it also implies success on instances in certain other equivalence classes. As suggested earlier, the equivalence classes associated with a given partition may be partially ordered as to difficulty.

The basis for this partial ordering, recall, resides in the nature of the corresponding paths. Thus, certain paths include others in the sense that the former contain all of the atomic rules of the latter, and in the same order, plus some additional ones. Consider, for example, the task used in experiments 12-15. In this case, paths 5, 4, and 3 were superordinate, respectively, to paths (4, 3, 2, 1), (1), and (2) as shown

```
  4 5 3
  1
  2
```

The data show that in all 12 cases where a subject was consistently successful on a superordinate path, he was also successful on the subordinate paths. The converse is not always true. Of the 8 cases where the subjects were successful on all of the subordinate tasks, they were also successful on the corresponding superordinate tasks 6 times.

In all, there were 205 cases in which one path was superordinate to another and in all but 7 or 3% of these cases success on a superordinate task implied success on the subordinate ones. The 95% confidence interval for the obtained 97% prediction level is between 95% and 99%.
It should be noted that 4 of the 7 exceptions can be attributed to the strong possibility in experiments 12-15 that two of the subjects generated a procedure from the given atomic rules which differed from that on which the analysis was based. This possibility could not have had an effect on the within-equivalence-class analysis because the same partition would have resulted. But, it could have affected the hierarchy analysis.

Conversely, subjects were consistently successful on superordinate tasks in 90 of 117 cases in which they were also successful on all subordinate tasks at the next lower level. These results are directly comparable to those obtained using task analysis, which was pioneered in education by Gagné (1962) and his collaborators (e.g., Gagné, Mayor, Garstens, and Paradise, 1962). There too, successful performance on superordinate tasks is almost always indicative of success on subordinate tasks, and success on all of the subordinates frequently (about 80% of the time) implies success on corresponding superordinate tasks.

This should not come as a surprise because the hierarchical algorithmic analysis proposed parallels task analysis directly. The major difference is that our intentional analysis allows for denumerably many different ways of solving a given class of tasks, whereas task analysis implicitly assumes that there is just one. Analysis in terms of procedures is also more precise and explicit in the sense that the analyst is forced to make his intuitions public. It is interesting to note that some task analysts have recently begun to move in this direction (Resnick, 1970).

As suggestive as they might be, our experimental results deal only indirectly with the problem of assessing the behavior potential of given subjects with respect to given classes of rule-governed behavior. In these experiments, the subjects were trained in the atomic rules. To assess behavior potential, the observer must make intelligent guesses as to which procedures a given subject (or class of subjects) might reasonably use, including judgments concerning the component (i.e., atomic) rules. The feasibility of this algorithmic approach was tested by Durnin and Scandura (1971) and the essentials are reported in the next section.

3.6 Assessing Behavior Potential

In addition to testing the feasibility of the algorithmic approach to assessing behavior potential, the Durnin and Scandura (1971) study also compared this approach with two others that have recently been proposed—item forms (Hively, Patterson & Page, 1968) and hierarchical analysis (Ferguson, 1969). In general, it was found that the algorithmic approach
was not only more exacting in an analytical sense but also roughly twice as efficient, requiring many fewer test items to achieve even higher levels of predictability.

The research reported in this section deals primarily with the algorithmic approach. For details concerning the comparison with the other methods see the full report.

**Method.** The algorithmic approach was used to construct four algorithms for column subtraction. Two algorithms were based on a "borrowing" procedure for subtraction and consisted of 6 and 5 paths, respectively. The other two algorithms were based on an "equal additions" procedure and consisted of 4 and 8 paths, respectively. The intersection partition with respect to all four algorithms was then constructed, and contained 12 equivalence classes. The flow chart of the subtraction algorithm shown below was designed explicitly to have a path corresponding to each and every equivalence class in the intersection partition.

---

15I am indebted to John Durnin for permission to include portions of our paper prior to its publication.
The directed graph, the twelve possible paths, and items from corresponding equivalence classes of the flow chart are shown in the figure below. The numbered arcs in the graph and paths correspond to operations in the flow chart and the points to START, STOP, and decision diamonds of the flow chart.

Directed Graph

<table>
<thead>
<tr>
<th>Paths</th>
<th>Stimulus Instances from Corresponding Equivalence Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. $1 \rightarrow 2 \rightarrow \ast$</td>
<td>7</td>
</tr>
<tr>
<td>2. $1 \rightarrow 4 \rightarrow \ast$</td>
<td>13</td>
</tr>
<tr>
<td>3. $1 \rightarrow 2 \rightarrow \ast$</td>
<td>258</td>
</tr>
<tr>
<td>4. $1 \rightarrow 2 \rightarrow 4 \rightarrow \ast$</td>
<td>153</td>
</tr>
<tr>
<td>5. $1 \rightarrow 2 \rightarrow \ast$</td>
<td>54</td>
</tr>
<tr>
<td>6. $1 \rightarrow 5 \rightarrow 4 \rightarrow \ast$</td>
<td>1563</td>
</tr>
<tr>
<td>7. $1 \rightarrow 5 \rightarrow 2 \rightarrow \ast$</td>
<td>268</td>
</tr>
<tr>
<td>8. $1 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow \ast$</td>
<td>1663</td>
</tr>
<tr>
<td>9. $1 \rightarrow 5 \rightarrow 2 \rightarrow \ast$</td>
<td>603</td>
</tr>
<tr>
<td>10. $1 \rightarrow 5 \rightarrow 3 \rightarrow \ast$</td>
<td>4029</td>
</tr>
<tr>
<td>11. $1 \rightarrow 3 \rightarrow 4 \rightarrow \ast$</td>
<td>1300</td>
</tr>
<tr>
<td>12. $1 \rightarrow 4 \rightarrow \ast$</td>
<td>16059</td>
</tr>
</tbody>
</table>

Figure 4
Prediction and criterion tests (parallel tests A and B respectively) were constructed by generating two arbitrary items for each of the 12 equivalence classes, one for each test. The order of items was randomized in each test.

The subjects were 34 ninth grade general mathematics students attending summer school at Shaw Junior High School in West Philadelphia. Tests A and B were administered to the subjects in their classrooms as a group on consecutive days. The order in which the tests were given was counterbalanced over subjects. Of the 34 subjects, 25 were in attendance both days and received both tests A and B.

**Results and Discussion.** The overall level of predictability from Test A to Test B was 82%, with 85% prediction based on Test A items on which a subject was successful and 71% prediction where subjects were not successful. The correlation was .53. Furthermore, tests formed from the two algorithms based on "borrowing," with only 6 and 5 items, respectively, had 65% and 75% levels of prediction where subjects were unsuccessful on Test A items, with overall levels of predictability at 78%.

It may be noted parenthetically that of the four algorithms originally identified, the two based on "borrowing" had significantly higher (p<.05) levels of prediction than the two algorithms based on "equal additions" where subjects were unsuccessful on Test A items (65% and 75% as compared to 29% and 32%). This difference appears to reflect the fact that "borrowing" is the more common procedure taught in American schools.

Other data strongly suggest that under the testing conditions used, the algorithmic approach for assessing mastery approaches asymptote. Further partitioning of the given rule-governed class led to essentially no further improvement in predictability, even with as many as 37 test items. Further improvement would almost necessarily require more rigorous testing conditions analogous to those reported in Section 3.5.

In addition, the efficiency of the approach can, in principle, be increased even more because the approach provides an explicit basis for...
ordering equivalence classes according to difficulty that is independent of empirical data. Although the fit was not perfect, the obtained data were compatible with the obtained ordering. In only 35 of 144 cases were subjects successful on a superordinate equivalence class but not successful on all of the subordinate ones.

To summarize, then, the algorithmic approach not only provides an efficient, near asymptotic method for assessing behavior potential, but also the hierarchy induced by the approach may reasonably be used to increase this efficiency even more through the use of conditional testing procedures which involve branching.

Further Comments. Clearly, the algorithmic approach makes it possible, theoretically at least, to conceive of highly efficient testing techniques. Such techniques would capitalize, not only on the identification of a finite number of equivalence classes, but also on the partial ordering imposed on these equivalence classes by the underlying procedures. According to the hierarchy, testing would begin with the most difficult items and stop with the first successes. Such techniques could play an important role in computer assisted or other forms of automated or semi-automated testing.

Once given an algorithm, for example, a computer could be programmed to automatically trace out the paths, identify the equivalence classes of problems, randomly generate test items in the equivalence classes, and order the items for testing. Moreover, on further reflection, it becomes apparent that the algorithmic approach has a further major advantage. It provides an explicit basis for individualized instruction. To see this, we assume in accordance with our theory (Scandura, 1971a, 1971b, 1972b) that subjects actually use rules (algorithms) to generate their behavior. Then, because each equivalence class of items corresponds to a unique path of a rule, and because the steps in each such path are known explicitly to the computer, each student can be given specific instruction to overcome his inadequacies. Put succinctly, he can be taught the needed paths. These ideas constitute the theoretical basis for a series of self-diagnostic and self-instructional tapes and workbooks on computational skills in arithmetic that have been developed by the Mathematics Education Research Group under my direction (cf. Scandura, 1970a).

A major practical advantage of this approach over traditional (stochastic) forms of testing is that it is self-correcting. If a particular analysis is in error, one knows exactly where to look for difficulties. When a probabilistic model is used, the test constructor simply knows how
adequate or inadequate the predictions are. There are no explicit guidelines as to where to look in order to improve the level of prediction.

In the algorithmic approach, poor prediction may result either because one or more underlying rules are overlooked in the analysis or because what are presumed to be atomic rules do not turn out to be atomic. In the latter case, for example, further refinement is called for. The class of behaviors associated with each presumed atomic rule can be analyzed in exactly the same way as with any other class of rule-governed behaviors; the process involved is analogous to analyzing a subroutine of a computer program. For example, consider Step 1 in the illustrative algorithm of Section 3.3 for generating next numerals in base three: "Read (encode) the last digit of a given (base three) numeral." Anyone who had mastered this rule would certainly be able to "read" the circled digits in 10®, 2®, 22®, 2012®, or in any other base three numeral. Although it would be difficult to find an adult for whom this ability did not act in atomic fashion, the situation with a young child just learning to read might be quite different. It might be necessary, for example, to replace step 1 by an equivalent procedure composed of less molar atomic rules. An obvious possibility might be to distinguish one-digit numerals from the rest. In this case, Step 1 might be represented

where one arc corresponds to reading one-digit numerals and the other to multi-digit numerals.\(^\text{17}\)

After introducing additional procedures and/or breaking down each presumed atomic rule into more suitable subatomic rules, the test constructor could then construct and try out a new test in the same manner as before. This procedure could be repeated any number of times, but practically speaking, it is unlikely that an experienced test constructor would be off by very many levels of analysis.

As described, the proposed approach applies only to given rule-governed classes and individual subjects. If one is concerned with a more diverse population of problems or with predicting the behavior of groups, as in ordinary testing, some modifications must be made.

\(^{17}\)Furthermore, it is always possible in principle to determine a priori how adequate the atomic assumptions are. Thus, one can always test directly to determine whether those rules, which are assumed to be atomic, are indeed atomic. In general, we simply look to see whether success on one test instance implies success on any other test instance, and similarly for failure. If so, the rule is said to be atomic; otherwise, not.
Given a diverse population of problems, it is unrealistic to expect to find any single procedure which can generate all of the responses (solutions). In this case, the first step is to identify one or more rule sets which both account for the given corpus and come fairly close to matching what the subject(s) knows. The test instances, then, would be selected for each procedure (rule) in very much the same way as before. Of course, the number of procedures involved could be quite large, especially where relatively few higher order rules are involved (cf. Chapters 4 and 5). Nonetheless, I suspect that by following some such approach we could eventually do as good a job in constructing standardized tests, say, as is presently accomplished by random sampling from hypothetical and ill-defined populations of problems.

Conceptually speaking, extension of the proposed approach from individuals to groups is straightforward. It is sufficient to note that, in general, with groups (1) more different procedures may have to be taken into account for each rule-governed class, and (2) the underlying procedures may have to be refined to a greater degree because the atomic assumptions must be commensurate with the prerequisites had by the weaker members of the group. The important point is that when in doubt one can always devise alternative procedures and/or analyze any component of a given procedure into smaller parts. The additional test items would tend to increase overall reliability.

The only major disadvantage of over-refining procedures and/or introducing additional procedures would be to make assessing behavior potential less efficient than it might otherwise be. Even this inefficiency, however, can be counteracted to some extent by capitalizing on item difficulty.

4. THE NATURE OF PROBLEM SOLVING—A MECHANISM FOR LEARNING

The simple performance hypothesis of Section 2 applies where a subject has learned rules immediately available for achieving his goal. What happens where there are no such rules? In this case the subject has a problem in the classical sense—a problem situation, a goal, and a barrier between them.18

The major theoretical problem is to explain what happens when a subject is confronted with such a situation. If the problem can be dealt with

18Although memory is an important factor in solving many kinds of problems, discussion of its general nature is postponed until Chapter 10. The theorizing involved in this and the following two chapters is concerned only with memory-free conditions. As such, all learned rules are assumed to be readily available to the subject.
in a way which lends itself to prediction, so much the better. Why certain people are able to solve problems for which they have never before learned a specific rule whereas others cannot, is the question of particular interest. We want to know exactly what is involved and why subjects perform as they do.

The general point of view adopted here is that rules underlie all behavior. In order to solve a problem, therefore, a subject must first acquire a solution rule. Indeed, according to the present analysis, the question of problem solving and the question of learning are essentially identical. Learning is a problem solving process and problem solving depends on learning (rules).

The major purpose of this section is (1) to propose a simple mechanism for learning, (2) to show how this mechanism, together with the performance mechanism of Section 2, can be used to explain relatively complex examples of problem solving, and (3) to describe empirical tests of this mechanism. In Sections 5 and 6, respectively, we show how these same mechanisms can be used to explain learning by discovery and learning by exposition. Under the proposed memory-free conditions, it is assumed throughout that once a given rule has been learned it is available from that point on.

As a first approximation, it appears that a very simple mechanism may suffice. This mechanism can be framed as a hypothesis as follows: Given a goal situation $(S,G)$, for which a subject does not have a learned rule immediately available, control reverts to the higher order goal $HG$, which may be satisfied by any procedure $p$ whose domain includes $S$ and whose range is contained in $G$. (We also say that the outputs generated by such a procedure satisfy $G$.)

With $HG$ in force, the subject presumably selects from among the available and relevant higher order rules in the same way as he would with any other goal. Furthermore, where no adequate higher order rules are available, one might suppose that control would revert to still higher order goals. Theoretically this process could continue indefinitely, but I suspect that a subject would tire of it (or run out of higher ... higher order rules) as quickly as would a theorist attempting to describe what was happening.

4.1 Basic Hypotheses and Examples

For present purposes, we restate our basic hypotheses as follows.
H₁ (Simple Performance): Given a subject in \( (S,G) \), where \{rᵢ | G may be satisfied by rᵢ for all i \} contains all of the relevant rules available to the subject, then the subject will apply some \( rᵢ \) to \( S \).

Similarly, the higher order hypothesis above can be restated

H₂ (Higher Order): Given \( (S,G) \) where \{rᵢ | G may be satisfied by \( rᵢ \) for all i \} = \( \emptyset \) (where \( \emptyset \) is the empty set), then control temporarily shifts to \( HG \).

To complete things, we need a third hypothesis which allows control to revert back to lower order goals once a higher order goal has been satisfied. To an approximation, this reversion hypothesis can be stated

H₃ (Reversion): Given that \( HG \) is satisfied, then control reverts back to \( G \), where \( HG \) is satisfied means that some procedure \( p \) has been derived which might satisfy \( G \) (i.e., such that \( S \in \text{Domain } p \) and \( \text{Range } p \subseteq G \)).

Although only implicit in what has been said, it is important to note that each of these hypotheses is assumed to apply at all levels. For example, \( H₁ \) applies in higher order goal situations as well as simple ones. (A completely rigorous formulation of the basic learning and performance mechanisms, unfortunately, is not easily achieved without more formal machinery and is postponed until Chapter 9. In order to fully appreciate the formal mechanism, the reader is urged to first consult Chapter 8 where the motivation mechanism is described.)

Suppose, for example, that a subject \( S \) is placed in goal situation \( (S,G) \) and has only the following rules available: \( r₁, r₂ \) (neither of which satisfy \( G \)), and \( h \) (where \( h(r₁,r₂) = p \) and \( p \) satisfies \( HG \)). What can we say about what the subject will or won't do? More particularly, given that a subject has adopted goal \( G \) in situation \( S \), and only has rules \( r₁, r₂ \), and \( h \) available, do the three hypotheses above provide a sufficient basis for predicting the subject's behavior? To see that they do, we observe

\[
\{rᵢ | G \text{ may be satisfied by } rᵢ \} = \emptyset \text{ (is empty)} \quad \text{Assumption}
\]

(1) \( \therefore \) Control shifts to \( HG \)

\( HG \) can be satisfied by \( h \)

(2) \( \therefore S \) will apply \( h \) to \( r₁, r₂ \) (to get \( p \))

\( p \) satisfies \( HG \)

(3) \( \therefore \) Control reverts back to \( G \)
p is available where G can be satisfied by p
(4) \[ S \text{ will apply } p \text{ to } S \text{ (to get } R) \]
R may satisfy G

If we changed the situation, say, by eliminating any one of \( r_1, r_2, \) or \( h, \) we could conclude that the subject would fail. (This conclusion, of course, would be relative to the adequacy of the assumption that the listed rules exhaust those which the subject has available for achieving the goal.)

A simple corollary of this observation provides a possible answer to the question of why some subjects succeed on problems for which they have learned all of the constituent parts whereas others do not. Apparently, the successful subjects have already mastered an appropriate higher order rule.\(^{19}\)

Consider some specific examples. First, suppose the problem (goal situation) posed to a subject is to convert 2 yards (S) into inches (G). In this case, there are two obvious ways in which a subject might solve the problem. The first is to simply know and have available a rule for converting yards directly into inches.

The other way is more interesting and involves all of the mechanisms described above. Here, we assume that the subject has mastered one rule \( r_1, \) for converting yards into feet, and another \( r_2, \) for converting feet into inches. The subject is also assumed to have mastered a higher order rule \( h, \) which allows him to generate compositions of learned rules. That is, \( h \) operates on pairs of rules in which the outputs of one match the inputs of the other, and \( h \) outputs new rules which have the same domains as the former rules and ranges corresponding to the latter rules. (See Chapter 5 for a more precise characterization.)

In a situation of this sort, the subject will adopt the higher order goal \( HG \) which can be stated: Derive a procedure \( p \) such that "2 yards" is in the domain of \( p \) and the outputs of \( p \) are a number of "inches." In the manner described above, the subject next selects \( h \) and applies it to the arguments \( r_1 \) and \( r_2, \) yielding the composite rule "Multiply the given number of yards by three and relabel with 'feet'; then multiply the output by 12 and relabel with 'inches'." Finally, control reverts to the original goal and the subject applies the newly derived rule to "2 yards" to generate

---

\(^{19}\)What has typically been called problem solving ability, according to this theory, resides in the availability of higher order rules. Hence, a "pure" test of problem solving ability should provide for factoring out direct solutions. All other things being equal, solving a problem without \textit{a priori} knowledge of a specific solution rule requires more problem solving ability than solving the problem with such a rule.
the desired response "72 inches."

Generally speaking, higher order rules act in many different kinds of problem situation. Thus, for example, the higher order rule $h$ may apply not only to situations where yards are being converted to inches but to parallel problems involving different kinds of measures—for example, problems in which the subject must combine rules for converting gallons into quarts and quarts into pints. Potentially, $h$ may be applied to any problem in which a pair of prelearned rules must be combined so that the output of one serves as the input of the other.

The second problem we shall consider is taken from Polya (1962) and involves constructing a circle about a given triangle $ABC$ with ruler and compass. This problem is somewhat more complex and illustrates how the analysis may be extended to still higher levels. More particularly, the problem may be solved in any one of at least three ways.

At the simplest level, the subject may simply apply (by $H_1$) a learned rule $p$. The first step in this rule, denoted $r_1$, makes it possible to construct the locus of points (the perpendicular bisector) equidistant from two vertices (say $A$ and $B$) of the triangle. The second step is to apply $r_1$ again to a different pair of vertices (say $B$ and $C$). The final step $r_2$ is to set the compass (radius) equal to the distance between the intersection of the two loci and any one of the vertices, and to construct a circle with this radius using the point of intersection as the center.

A subject might also succeed even where he had not explicitly learned rule $p$. Suppose, for example, that the subject has rules $r_2$ and $h$ available, together with the ability to find points equidistant from three given points, denoted $r$. Under these conditions, the subject might reasonably be expected to solve the problem as before. That is, control would first revert to the higher order goal of deriving a procedure which works. The subject would then apply rule $h$ to $r$ and $r_2$ to generate $p$. Finally, control would revert to the original goal and the subject would apply $p$ to get the correct answer.

Interestingly enough, this does not appear to parallel very well what Polya (1962, 5) would have subjects do. In attacking problems of this type, he suggests: First, reduce the problem to the construction of a single point—for example, the center of the to-be-circumscribed circle. Then, split the (goal) condition to be satisfied into two parts—for example, consider the center as the intersection of two loci.

Anyone who has read Polya's works cannot help but be impressed with the wealth of examples he has used and with his keen insights into problem solving. It would seem almost essential that any serious attempt to deal
systematically with the mechanisms of problem solving make contact with Polya's ideas.

In the present context, the key questions appear to be: (1) What role in our framework do heuristics play? (2) What (other?) specific abilities does a subject need in order to succeed on the given problem? These questions can be answered but only by introducing a third level of analysis. More specifically, we shall assume that the following rules are available to the subject.

(a) \( r_2 \) (above)
(b) \( r_1 \) (above)
(c) \( h_1 \), a rule for forming conjunctions of given rules in which the conjunctions involve first applying a given rule to one part of a stimulus and then to another part (e.g., the conjunction "Apply \( r_1 \) to vertices A and B and then apply it to vertices B and C"—this corresponds to combining loci, each of which satisfies one condition.)
(d) \( h_2 \), the ability to form a composite of two given rules. \( h_2 \) enters into our analysis at two levels. At one level it corresponds to "reducing" the problem to that of finding one point and of deriving a procedure for finding that point. At the other level it serves as part of a procedure \( h' \) for deriving the solution procedure \( p \).

Throughout this analysis, we assume the necessary decision making capabilities. (For more details see the description of the problem solving experiment in Section 4.2. Also see Chapter 9.)

Given that these rules are available, and letting \( (S,G) \) denote the goal situation, Polya's solution method can be characterized as follows.

\[
\{ r_1 \mid G \text{ may be satisfied by } r_1 \} = \emptyset \text{ so control shifts to } HG.
\]
\[
\{ h_1 \mid HG \text{ may be satisfied by } h_1 \} = \emptyset \text{ so control shifts to } H^2G.
\]
\[
\{ h_1 \mid H^2G \text{ may be satisfied by } h_1 \} = h_2 \text{ (where } h_2 \text{ corresponds here to the heuristic "reduce the problem to that of finding one point and, then, find that point")}.
\]

Hence, the subject will apply \( h_2 \) to \( h_1 \) and \( h_2 \) (itself) to get \( h' = h_1 \) followed by \( h_2 \).

\( h' \) satisfies \( H^2G \) so control reverts back to \( HG \).

But \( HG \) may be satisfied by \( h' \).

---

\[\text{In the composites so formed the second rules, strictly speaking, may not operate on outputs of the first rules. Rather, they operate on these outputs together with other inputs in the rule set.}\]
Hence, the subject will apply \( h' \) to \( r_1, r_2 \) to get \( h'(r_1, r_2) = p \).

\( p \) satisfies \( HG \) so control reverts back to \( G \).

But \( G \) may be satisfied by \( p \).

Hence, the subject will apply \( p \) to \( S \) to get \( R \). \(^{21}\)

To summarize, it would appear that given appropriate higher order rules (heuristics), subjects may succeed on tasks even when they do not have a learned solution rule immediately available.

It is important to emphasize, however, that a given higher order rule, or heuristic, will not always lead to a solution. In the first place, heuristics typically have a limited range of applicability. The available ones simply may not be relevant. Second, the subject may not have all of the required prerequisite rules available.

A third, more subtle reason is that there is no guarantee that the rules derived at any given stage of a derivation will automatically, according to the above mechanisms, lead to a solution. A newly generated rule might satisfy a higher order goal, for example, but when applied to the stimulus it may not generate a response which satisfies the original goal. Further discussion of this important matter, and a possible resolution, is postponed until Chapter 9.

If borne out in empirical research, the proposed form of analysis could have important implications both for research in artificial intelligence and for education.

Learning in artificial intelligence systems, for example, has general-

---

\(^{21}\)It should be emphasized that this analysis ignores many details of the derivation process, particularly regarding the exact nature of the rules. For example, nothing was said in any of the above analyses concerning the scope or generality of the given rules. This is an important limitation, particularly as this involves such (potentially) highly general rules as *(c) and *(d) above. Almost from the time a child is born, for example, he frequently seems able, say, to form the composite of certain learned rules. Once a baby is able to get a bottle into its mouth and is also able to "prop it up" once it is put in its mouth, for example, it would not be surprising to see the baby perform the composite act.

According to the present analysis, subjects learn to increase the scope of such rules gradually via application of generalizing rules of the sort discussed in Chapters 5 and 6. Generalizing rules may play an important role in learning a wide variety of common and potentially highly general rules. Their role in learning logical rules of inference may be particularly crucial.

As this book goes to press, substantial progress is being made in analyzing in detail realistic prototype instances of mathematical problem solving, in accordance with the formal theory described in Chapter 9. This research is being supported by the National Science Foundation and a report will be issued on its completion.
ly been nonexistent, or based on some form of conditioning theory (cf. Simon & Newell, Volume II; Minsky, 1968). Even where the power of an existing system can be greatly increased upon presentation of limited items of additional information (cf. Bobrow in Minsky, 1968), the way in which this is accomplished and how individual programs (rules) interact varies from system to system.

The proposed mechanisms would provide a fixed mode of interaction for the rules characterizing each (artificial intelligence) system—a mode of interaction designed to reflect human behavior. That is, these mechanisms would provide constraints on the traditionally free-wheeling atheoretical approaches which have characterized most of the research to date in this area (cf. Nilsson, 1971). In effect, the proposed theory could provide a possible foundation upon which artificial intelligence researchers may build. It is too early to make any firm statements concerning the future value of such research, but, as the following experiments suggest, it could help bring about better coordination between the fields of psychology and artificial intelligence.

The proposed theory also suggests new approaches to individualized instruction, based on operational objectives (e.g., Lipson, 1967) and task analysis (e.g., Gagné, 1962). In order to guarantee desired terminal performance, for example, the so-called prerequisites alone are not sufficient. The subject must also be given directions as to how these prerequisites are to be combined (Gagné, 1962). Still, some subjects are able to produce the necessary integration without directions. Why? The proposed analyses suggest that such subjects are successful precisely because they have already mastered appropriate higher order rules. If we took the trouble to determine which higher order rules subjects have available, we could make learning more efficient for many students; and for others, we might well overcome otherwise intractable problems.

Similarly, in teaching children how to solve problems, we typically concentrate on explicit procedures for solving particular classes of problems. Few serious attempts have been made to even identify what might be called heuristics or higher order procedures. Progress in this direction could lead to qualitative improvement in education. Teaching higher order rules to students might well make it possible for them to derive procedures on their own for solving many different classes of problems.  

22 Some progress has recently been made in developing instructional materials to accomplish this objective (Scandura, Durnin, Ehrenpreis, & Luger, 1971), and the results of an empirical test were most encouraging (Ehrenpreis & Scandura, 1972).
In order to make predictions in actual instances of problem solving, or course, it is essential to know what rules a subject actually knows; and as we saw in Section 3, this involves assessing behavior potential. The interdependence of theorizing about competence and theorizing about behavior, therefore, becomes evident once again. Because such assessment always involves a certain degree of uncertainty (i.e., is dependent on the adequacy of the presumed theory of competence), testing the basic mechanism itself empirically can be accomplished most directly by experimental manipulation of the assumed rules.

4.2 Experiments on Problem Solving

The proposed analyses seem to have a good deal of intuitive appeal, but it is quite possible that the proposed mechanism might not be reflected in actual behavior (under memory-free conditions). Unfortunately, most of the available problem solving research (e.g., Bartlett, 1932; Dunker, 1945; Wertheimer, 1945) is at best suggestive on this point. Perhaps the most relevant experiment in the literature is one of my own (Scandura, 1968; see Chapter 2, Volume II) and, even here, the experiment was designed with something else in mind. In this study, all of the subjects were taught to use a higher order rule (use of parentheses). Half were also taught one or more (lower order) rules (e.g., to determine the greatest integer in a number, denoted \([x] \rightarrow n\)). The subjects who learned the lower order rules were able with 80% certainty to combine them via parentheses to form new composite rules (e.g., \([([x] + [y])] \rightarrow m\)). The other subjects were uniformly unable to do so.

EXPERIMENT I - COMPOSITION

Louis Ackler and I recently completed a direct test of the mechanism.

Method

**Tasks and Materials.** The experimental material consisted of a set of tasks involving trading stimulus objects of one kind (e.g., red chips) for response objects of another (e.g., pencils). Each such task can be characterized as a set of stimulus-response pairs in which \(n\) stimulus objects are mapped into \((n+m)\) response objects. One task, for example, involved mapping \(n\) red chips into \(n+3\) pencils (i.e., 2 red chips into 5 pencils, 4 red chips into 7 pencils, and so on). Throughout the experiment, \(n+m \leq 10\), with \(n \leq 7\) and \(m \leq 4\).

There were two kinds of rules, *simple* and *composite*, for solving such
tasks. Simple rules effected the trades directly and were represented on 5" x 8" cards. The card illustrated at the top of Figure 5 represents a simple rule which maps n paper clips into n+1 blue chips. On the actual card, a paper clip was glued on the left and a blue chip on the right.

Figure 5

The second kind of rule is illustrated similarly at the bottom of Figure 5. The composite rule represented first changes pencils (unsharpened) into paper clips and then paper clips into white chips. The goal of each experimental task, then, was to effect the given type of trade by applying a rule of one of these two kinds.

A pair of simple rules is said to be compatible if the outputs of one of the rules are of the same type as the inputs of the other. Clearly, compatible rules can be combined to form composite rules. For example, the rules denoted "n caramels \( \rightarrow \) n + 1 toy soldiers" and "n toy soldiers \( \rightarrow \) n + 2 pencils" can be combined to form a composite rule which maps n caramels into n + 3 pencils. The set of compatible pairs of simple rules comprises the domain of a higher order rule which maps such pairs into corresponding composite rules.

This higher order rule was used to define a second, higher order kind of task in which the goal is to devise a composite rule for effecting a given kind of trade. For example, the goal might be to construct a composite rule for converting caramels into pencils given the rules denoted "n caramels \( \rightarrow \) n + 2 white chips" and "n white chips \( \rightarrow \) n + 1 pencils."
Subjects, Design, and Procedures. The subjects were 31 elementary school boys (B) and girls (G) between the ages of 5 and 9. They were trained and tested individually and were given twenty five cents for participating.

At the start of the experiment, each subject was told that he was going to play a trading game with the experimenter and was given several sets of objects to trade. The subject was then taught to interpret the rule cards and to make trades using the rules represented by the cards. For example, the subject was shown a card like the one at the top of Figure 5 and was told, "Here is the rule we will use to trade paper clips for blue chips. It says, no matter how many paper clips I give you, you must give me the same number of blue chips and then add one more. Now, here are three paper clips. How many blue chips should you give to me?" If necessary, the experimenter showed the subject how to make a trade of this sort and asked him to repeat what he had been shown.

The experimenter initiated a number of trades requiring use of the rule, providing assistance where necessary until the subject reached a criterion of three consecutive successful trades. The experimenter then gave the subject a set of objects not in the domain of the rule, say, 2 pencils, and asked the subject, "Can you use this rule to trade these pencils for blue chips?" Regardless of the subject's response, the experimenter went on to emphasize that the rule could be used only to trade paper clips for blue chips.

The experimenter then showed the subject a different rule card and asked him to interpret the rule, providing assistance if necessary. Practice at using the rule to make trades then followed and continued until the subject reached criterion. Again, it was emphasized that the rule could only be used to trade the objects appearing on the card. This procedure was repeated using different rules until the subject was able to interpret rule cards and apply the corresponding rules without assistance for three different, consecutive rules. At this point, it was assumed that for the subject, simply seeing a rule card was equivalent to knowing and being able to apply the rule.

During the second phase, the subject was taught to interpret and use the composite rules. The subject was told, in the case of the rule shown in Figure 5: "Here is a rule for trading pencils for white chips. Let me show you how it works. Suppose I give you two pencils. Use the first rule (the experimenter pointed at the first card) to trade these pencils for paper clips. How many paper clips will this give?" The subject was
required to place the correct number of paper clips on the table. "Now, use the second rule (again pointing) to trade these four paper clips for white chips. How many white chips should you give me for four paper clips?" This procedure was repeated with different stimuli until the subject performed three consecutive trades correctly. Practice then continued with similar rules until the subject interpreted and correctly applied three consecutive composite rules.

The subject was then given a pretest which consisted of two parts. First, the subject was presented with cards representing a pair of compatible rules. The subject had not seen either rule card before, but it was assumed that, by virtue of his earlier training, he knew what the cards meant. Then, the subject was asked to make three trades requiring use of the corresponding composite rule. (He was never shown this rule directly, either before or after testing.)

For example, a subject who was presented with the rules "n pencils → n + 2 pieces of bubble gum" and "n pieces of bubble gum → n + 1 paper clips" would be presented in turn with various numbers of pencils (e.g., 2, 4, and 1) and asked to trade the appropriate numbers (e.g., 5, 7, and 4) of paper clips. No reinforcement was provided on the pretest.

Those subjects who consistently failed on the pretest were randomly assigned, in matched pairs, to one of two treatment groups, Group HR, the group which received training on the higher order rule, and Group C, the control group. Those subjects who succeeded on two or more instances of the pretest were given 5 minutes of irrelevant instruction (reading a comic book) and were given a posttest, which involved new rules but paralleled the pretest in every other respect.

Each subject in Group HR was taught the higher order rule. To accomplish this, the subject was first shown two compatible rule cards and was asked to interpret each. The experimenter then demonstrated how the rules could be combined by sliding the rule cards together in the appropriate manner. The subject was then asked to interpret this newly formed rule. The experimenter emphasized that the rules could be combined only because the output of one was the same as the input of the other.

Following this, the subject was presented with several pairs of rules, some of which were not compatible. For each pair, the subject was required to form the composite rule, if possible, and to interpret the newly formed rule. For the randomly interspersed incompatible pairs, the subject had to indicate that the rules could not be so combined. After performing successfully on five consecutive pairs of rules (including both compatible
and incompatible pairs), the subject was given the posttest. The time required to train the subject on the higher order rule was recorded.

Instead of higher order rule training, each Group C subject was asked to read a comic book for the same amount of time as his matched partner in Group HR had required to learn the higher order rule.

The posttest was then administered in the same manner as the pretest to all subjects.

Results and Discussion

Of the 31 subjects tested, 24 were included in the experimental comparison. Six of the other seven were disqualified because they passed the pretest; they solved all of the transfer problems. The seventh subject was assigned to the HR group but failed to learn how to interpret the composite rule cards.

The experimental results were relatively clear-cut. All but one of the 12 experimental (HR) subjects solved all three transfer problems. There was reason to believe that the one subject who failed after reaching criterion on the higher order rule may have forgotten what he had just learned. He required an unusual amount of help from the experimenter during higher order rule training. For this reason he was put through the experimental procedure a week later, and this time he performed perfectly on the transfer problems. All 12 of the control subjects failed uniformly on the transfer problems.

The individual results are summarized in Table 2. In the table, "+" indicates that the subject reached criterion and "-" that he failed to do so. The results of the readministration of the HR treatment to the one discrepant case are indicated in parentheses. Time on the experimental instruction is reported in minutes and seconds. Subjects are identified according to age (5, 7, 8, 9) and sex (B, G).

Although statistical comparisons between the groups seem inappropriate, useful evidence concerning the reliability of the percentage of correct predictions may be expressed in terms of confidence intervals. This is only possible, of course, in the experimental treatment where the original predictions were not perfect. Based on the assumption that correct and incorrect predictions are binomially distributed and using the obtained mean percentage of 92% to estimate the expected percentage of correct predictions in Group HR, the 68% confidence interval was between 84% and 99.5%.
### TABLE 2
Summary of Problem Solving Results

<table>
<thead>
<tr>
<th></th>
<th>Experimental (HR) Subjects</th>
<th>Control Subjects</th>
<th>Disqualified Subjects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpreting single rule cards</td>
<td>+ + + + + + + + + + + + + + +</td>
<td>+ + + + + + + + + + + + + +</td>
<td>+ + + + + + + + + + + + + + +</td>
</tr>
<tr>
<td>Interpreting composite rule cards</td>
<td>+ + + + + + + + + + + + + + +</td>
<td>+ + + + + + + + + + + + + +</td>
<td>+ + + + + + + + + + + + + + +</td>
</tr>
<tr>
<td>Pretest</td>
<td>- - - - - - - - - - - - - - -</td>
<td>- - - - - - - - - - - - - - -</td>
<td>- - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>Time on Experimental (HR or irrelevant) instruction</td>
<td>6:10 4:50 3:45 3:45 4:30 4:00 5:05 3:45</td>
<td>6:10 4:50 3:45 3:45 4:30 4:00 5:05 3:45</td>
<td>5:00 5:00 5:00 5:00 5:00 5:00 5:00 5:00</td>
</tr>
<tr>
<td>Transfer (Re)test</td>
<td>+ + + - + + + + + + +</td>
<td>- - - - - - - - - - - - -</td>
<td>+ + + + + + + + + + + + +</td>
</tr>
</tbody>
</table>

Note: Times in minutes and seconds.
EXPERIMENT II - GENERALIZATION

A second experiment was conducted in collaboration with John Durnin and Francine Endicott. The purpose of this experiment was to test the mechanism under perhaps more demanding circumstances, with two different and more complex higher order rules of varying generality. Both involved generalization from restricted to more general rules.

The report below was purposely written to parallel that of Experiment I in order to accentuate the similarities and differences.

Method

Tasks and Materials. The experimental tasks involved responding with appropriate numerals to given stimulus numerals. Each task considered can be characterized as a set of stimulus-response pairs in which each number \( n \) is mapped into a number of the form \( an + d \), where \( a \) and \( d \) are whole numbers. For example, one task involved mapping \( n \) into \( 4n + 13 \). In addition to being of the above form, the domains of the restricted tasks contained exactly three elements, the number 1 plus two consecutive numbers (e.g., 8 and 9).

Restricted tasks were represented as triples which also designated the underlying rules. The triples were always presented to the subjects together with the three input numbers; the pairs were printed on the left of an 8 1/2" x 11" sheet of paper and the inputs on the right. A sample page is shown below.

\[
\begin{align*}
1 & \rightarrow 17 \\
8 & \rightarrow 45 \\
9 & \rightarrow 49
\end{align*}
\]

The rules underlying the unrestricted tasks were of two types. Type one \((an)\) rules simply involved multiplying the input numbers by \( a \) (i.e., \( d = 0 \)). Type two \((an+d)\) rules involved both multiplication and addition (of \( d \)).

Two higher order generalization algorithms were identified. Both higher order rules act on restricted rules (triples) and generate output rules of types one and two, respectively. The flow diagrams for the (higher order) Division Procedure and for the (higher order) Finite Differences Procedure, respectively, are represented in Figures 6 and 7. Both procedures partition the class of restricted rules into two equivalence classes consisting of \( an \) rules and \( an + d \) rules, respectively. The Division Procedure only generates \( an \) rules, whereas the Finite Differences Procedure generates \( an + d \) rules. (The Division Procedure may also be
Figure 6. Flow diagram for the higher order division rule which acts on restricted rules and generates general rules of the form \( n \to an \).

Figure 7. Flow diagram for the higher order finite differences rule which acts on restricted rules and generates general rules of the form \( n \to an + d \).
viewed as having a limited domain consisting of only those restricted rules in which the output of 1 equals the quotient of the output of one of the other pairs divided by its input.)

All practice sheets and test booklets were reproduced by ditto on 8 1/2" x 11" paper.

*Subjects, Design, and Procedure.* The 80 subjects were students in grades 5 to 12 of Catholic and public schools of Philadelphia. Eight subjects were unable to do simple arithmetic computations and hence were unable to participate. Twelve subjects who were able to solve $an$ problems but not $an + d$ problems on the pretest (phase 3 below) were also dropped to keep the experimental groups balanced. Each subject was trained and tested individually.

As in the previous experiment, Experiment II was run in five phases: (1) training in interpretation of restricted rules, (2) training in interpretation of general rules, (3) pretest on criterion problems, (4) training on a higher order generalization rule (or control), and (5) posttest on (new) criterion problems.

During the first phase, the subject was presented with triples of number pairs and shown how to interpret them. He was told that he could determine the appropriate output for a given input number in a triple by simply looking at the triple to see what output number was paired with it. The subject was instructed "to write the output number that goes with each input number according to the triple on the page." He was required to solve independently two tasks in succession before moving to the next phase.

During phase (2), the experimenter introduced general rules of the form $n \rightarrow an + d$, gave one specific example of a general rule where $d = 0$ and one where $d > 0$, and stated that for every triple of number pairs there is a general rule of the form $n \rightarrow an + d$ which "fits" the triple. The subject was told: "A general rule fits a triple if, when you use the general rule on the input numbers which appear in the triple to compute their output numbers, you get the same output numbers as in the triple." The subject was taught how to check whether a general rule fits a triple and practiced checking with four different rules. Two of these four rules were of the type $an + d$ and two, of the type $an$; one general rule of each type fit the triple accompanying the general rule. Then the subject was presented in turn with six pages, on each of which was a general rule, a triple of number pairs, and two to four input numbers not in the triple. He was told to "Find out what output number goes with each new input.
This involved first checking to determine whether the general rule fit the triple, and then, if it did, to use it to compute the output number for each new input number. The criterion was four consecutive correct problems worked independently.\(^{23}\)

Immediately prior to the pretest the experimenter summarized for the subject what he had learned. It was pointed out that there are two ways to find the output number for a given input number which appears in a given triple. One is simply to look at the triple to see what the output number for that input number is. The other way is to use a general rule, which fits the triple, to compute the output number for that input number. The subject also was reminded that he knew how to check whether a general rule fit a triple; and that, given a triple and a general rule that fits the triple, the output numbers for new input numbers can be computed via the rule.

The subject was then given a pretest consisting of four transfer problems, two of type \(an\) and two of type \(an+d\), presented in random order. Each transfer problem consisted of a triple and three new input numbers. The subject was instructed to "Find out what output number goes with each new input number according to a rule ... which fits the triple." If a subject solved one but not both of a type of problem (\(an\) or \(an+d\)), then he was given another problem of that type. If he solved it, he was given the unsolved test problem of that type to try again. Under these conditions, if he solved the original two problems of a type he was considered to have the ability to solve that type in general, otherwise not. When the subject encountered difficulty on the pretest, the experimenter explained that the task was very difficult. He encouraged the subject to try his best, without letting the subject become demoralized by his inability. Based on the pretest results, the subjects were categorized into three classes:

- **None** - The set of subjects who were unable to solve either type of problem,
- **\(an\)** - The set of subjects who could solve \(an\) problems but not \(an+d\) problems,
- **\(an+d\)** - The set of subjects who could solve both \(an\) and \(an+d\) problems.

The subjects in class **None** were randomly assigned to three treatment

\(^{23}\)All of the general rules used fit their accompanying triples. This was unintended, but it is very unlikely to have had any important effect on the results.
groups:

Control (C) - The group of subjects who received no higher order rule training,

Division (D) - The group of subjects who were taught the Division Procedure,

Finite Differences (F) - The group of subjects who were taught the Finite Differences Procedure.

Those subjects in class an were randomly assigned to treatment groups D and F; those in class an+d effectively knew all that we proposed to teach them and were unassigned.

Each subject assigned to either group D or group F was told, "Now I will teach you a procedure which can be used for some kinds of triples to find a general rule ... which fits the triple." On each higher order task the subject was presented with a triple and required to construct a general rule which fit the triple. The experimenter placed the appropriate higher order procedure, which was written out, on a stand in front of the subject and worked through two higher order practice tasks with the subject according to the steps in the procedure. Then the subject was presented in turn with six more higher order tasks and was required to reach a criterion of four correct solutions in a row, independently. The subjects assigned to group C and those in class an+d, who had solved all of the pretest problems, received no treatment.

Immediately prior to the posttest the experimenter again summarized what the subject had learned up to that time. This summary was identical to that given before the pretest, except for the subjects in the higher order treatment groups. For these subjects the following was added: "You also know a procedure which can be used for some kinds of triples to find a general rule ... which fits the triple. You may refer to this procedure whenever you wish." The procedure remained on the stand in front of the subject during the posttest which followed.

The posttest involved new transfer problems but paralleled the pretest in every respect.

Results and Discussion

Of the 60 subjects admitted, 50 were placed in the five experimental groups. 10 of the (60) subjects who received no treatment passed both the an and an+d type problems on the pretest.

The experimental results for all 50 experimental subjects are summarized in Table 3, and are exactly as predicted. None of the subjects in
# Table 3

Summary of Generalization Results

Number of Students Who Reached Criterion

<table>
<thead>
<tr>
<th>Groups</th>
<th>Restricted Rules</th>
<th>General Rules</th>
<th>(\alpha n) Problems</th>
<th>(\alpha n+d) Problems</th>
<th>H.O. Rule Treatment</th>
<th>Posttest (\alpha n) Problems</th>
<th>Posttest (\alpha n+d) Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>F</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>
group C of Class None were successful on either the an or an+d problems on the posttest. All ten subjects in the two D groups of Class None and Class an were successful on the an problems of the posttest but were uniformly unable to solve the an+d problems. All subjects in group F of both Classes were successful on the an+d problems. Needless to say, the predicted differences between treatment groups C, D, and F on both the an and an+d problems were all statistically significant (p < .001).

**EXPERIMENTS III AND IV - GENERALIZATION ASSESSMENT**

A third experiment was conducted, with Durnin and Wallace Wulfeck, to determine the feasibility of predicting the performance of individual subjects on specific problems by assessing the subjects' behavior potential. That is, we wanted to know whether or not the identified higher order rules provided an adequate instrument against which to measure the higher order knowledge had by individual subjects in the given population. Specifically, we used the methods of Section 3 to determine which parts of the higher order rules used in Experiment II were available to each subject. Predictions concerning each subject's performance on new problems were made accordingly, and then tested. Given the validity of the basic mechanisms, this experiment provides an indirect test of the compatibility of the identified higher order rules with actual human knowledge.

**Method**

The tasks and rules were the same as in Experiment II.

The materials were arranged into six 8 1/2" x 11" booklets. Booklets 1 to 3 covered the same material as Phases 1 and 2 of Experiment II. Material corresponding to Phase 3, the pretest on criterion problems, and Phase 4, training on higher order rules, were eliminated. Booklets 4 and 5 were inserted in its place. The purpose of these booklets was to make it possible to assess behavior potential with respect to the higher order rules. Because the outputs of the higher order rules were themselves (general) rules, booklet 4 was included to provide instruction and practice in how to write general rules of the form \( n \rightarrow an + d \). Page one read, "Suppose we want to write a general rule for a given triple in which we multiply the input number by 3 and add 4 to the product. First let \( n \) be the input number. To multiply \( n \) by 3 we write \( 3 \times n \). Then to add 4 to the product we write \( 3 \times n + 4 \). Therefore, we write the general rule \( n \rightarrow 3 \times n + 4 \)." Eight practice problems were given on pages 2 and 3.
Booklet 5 was used to test for the higher order rules, and read in part, "On each page of this booklet is a triple of number pairs. Find and write at the top of each page a general rule of the form \( n \rightarrow an + d \) which fits the triple on that page." Four problems followed. Two problems required use of a higher order rule for deriving general rules of the form \( n \rightarrow an \) and two problems, for deriving general rules of the form \( n \rightarrow an + d \). Booklet 6 corresponded to Phase 6, the posttest on criterion problems.

Experiment III was run with 17 first-year algebra students and Experiment IV, with 9 general math and 11 algebra students, all from West Philadelphia.\(^{24}\)

In Experiment III, booklets 1 through 6 were administered to the students in their classroom. One hour was allotted for the experiment. In Experiment IV, booklets 1 to 4 were administered on one day, and a short review booklet and test booklets 5 and 6 on the next. The class periods on both days were 40 minutes long.

The only major procedural change from Experiment II was that the instructions were read to classes instead of to individuals. Each subject was required to work at least four problems correctly in each training booklet. There were from 3 to 5 proctors available in each classroom to help the students.

In scoring test booklets 5 and 6, the problems were divided into two categories: (1) those involving a general rule of the form \( an \) and (2) those involving a general rule of the form \( an + d \). If a subject got both \( an \) problems correct in a booklet, then he was considered successful on that category; otherwise he was not. The same criterion was used with the \( an + d \) problems.

Although tests 5 and 6 both involved higher order rules, the difference between them is critical to the experiment and should be kept in mind. Predictions were based on the availability of appropriate higher order (and other) rules. A subject who could derive \( an \), but not \( an + d \), rules on test booklet 5, for example, was assumed able to solve transfer problems involving \( an \) rules whereas failure was predicted on the \( an + d \) transfer problems.

Results

The results of Experiments III and IV are summarized in Table 4

\(^{24}\)Thanks are due Dr. Katz, Mr. Watts and Mr. Mason for their cooperation in providing subjects, and Mrs. Baker, Mrs. Watkins, Mr. Karabinos and Miss Triman, who helped proctor the experimental sessions.
<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>No. Successful on Assessment Test (5)</th>
<th>Proportion Correct Predictions on Transfer Test (6)</th>
<th>No. of Ss not Successful on Assessment Test (5)</th>
<th>Proportion Correct Predictions on Transfer Test (6)</th>
<th>Total No. of Correct Predictions</th>
<th>Overall Percent Correct Predictions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EXPERIMENT III</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>an problems</td>
<td>17</td>
<td>7</td>
<td>4/7</td>
<td>10</td>
<td>10/10</td>
<td>14</td>
<td>82%</td>
</tr>
<tr>
<td>an + d problems</td>
<td>17</td>
<td>2</td>
<td>2/2</td>
<td>15</td>
<td>13/15</td>
<td>15</td>
<td>88%</td>
</tr>
<tr>
<td><strong>EXPERIMENT IV</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>an problems</td>
<td>20</td>
<td>13</td>
<td>11/13</td>
<td>7</td>
<td>7/7</td>
<td>18</td>
<td>90%</td>
</tr>
<tr>
<td>an + d problems</td>
<td>20</td>
<td>3</td>
<td>2/3</td>
<td>17</td>
<td>17/17</td>
<td>19</td>
<td>95%</td>
</tr>
<tr>
<td><strong>COMBINED</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>an problems</td>
<td>37</td>
<td>20</td>
<td>15/20</td>
<td>17</td>
<td>17/17</td>
<td>32</td>
<td>86%</td>
</tr>
<tr>
<td>an + d problems</td>
<td>37</td>
<td>5</td>
<td>4/5</td>
<td>32</td>
<td>30/32</td>
<td>34</td>
<td>92%</td>
</tr>
</tbody>
</table>

Table 4
and provide strong support for the theory.

The reliability of these results was tested with exact probabilities (Finney, 1948). In all cases, the number of correct, as opposed to incorrect, predictions differed significantly from chance ($p < .05$ for both $an$ and $an + d$ problems in Experiment III; $p < .001$ and $p < .05$, respectively, for $an$ and $an + d$ problems in Experiment IV; and $p < .001$ and $p < .005$, respectively, for $an$ and $an + d$ problems in the combined study).

Although the difference was not statistically significant, the additional 20 minutes provided in Experiment IV appeared to increase the precision of prediction on both $an$ and $an + d$ problems ($8\%$ and $7\%$, respectively). Whether or not the level of prediction could be further increased is not clear, but this observation does tend to support the notion that the results of such experiments may be expected to conform to prediction just to the extent that the required memory-free conditions are realized.

**GENERAL DISCUSSION**

Overall, the results of the first two experiments provide strong support for the postulated mechanisms. Under the experimental conditions, in which the effects of memory were minimized, the availability of an appropriate higher order rule appeared to be both a necessary and sufficient condition for solving the transfer problems. However, although we did not emphasize its importance, it should be noted that the ability to determine whether or not a given or derived rule satisfies a higher order goal is also crucial. This can be seen more clearly in the formalization of Chapter 9.

The results of Experiments III and IV put everything together. They show how a competence theory may interact with the proposed method of assessing behavior potential, and the hypothesized learning mechanisms, to provide a basis for explanation and prediction of behavior. Given a class of input-output pairs, the first task is to devise a finite set of rules, including higher order rules, which accounts for this class. The rules identified in Experiment II amount to doing just this. The next step is to determine which parts of which rules in the rule set are available to individual subjects. Test booklet 5 served this purpose. Finally, the mechanisms provide a basis for generating predictions concerning behavior. These predictions, recall, are based on the rules actually available to the subject, not those introduced by the experimenter, qua competence theorist.

Clearly, these experiments constitute only a beginning. The mechanisms
Mechanisms of Learning and Performance

themselves need to be further tested with other rules, and at varying levels of complexity. Indeed, as will become apparent in Chapter 9, there are fine-grained questions concerning the mechanisms themselves with which these experiments do not deal. Insofar as Experiments III and IV are concerned, we have hardly begun to empirically explore the limits and limitations of the approach.

APPLICATIONS

Although the amount of empirical evidence bearing on the proposed hypotheses is limited, this evidence does seem highly promising. Furthermore, the underlying mechanism is intrinsically simple, and appears to have considerable intuitive appeal.

In the following two sections, we shall see that these hypotheses provide a basis for analyzing complex human learning of a sort which to date has largely defied any attempt at detailed analysis. Section 5 is concerned with discovery learning, or learning from instances. Among other things, we show that the effective goal on any trial depends in a direct way on instances that have already been presented. In Section 6, we consider expository learning, or learning which takes place by extracting "meaning" from new statements in a language. Particular attention is given to the distinction between the ways in which symbols and icons denote (meaning).

The reader should not expect a completely definitive account, however. In the relative absence of data, much of what is said is necessarily speculative.

5. DISCOVERY LEARNING

The basic problem in discovery learning is to come up with a rule that accounts for a given rule-governed class on the basis of a finite set of S-R instances representative of that class.\(^{25}\) Since the number of S-R instances necessarily increases over trials, what are possible solution rules at one point in the discovery process may not be at another point. The goal conditions that must be satisfied on any given trial depend not only on the goal, as originally set forth, but also on the preceding S-R instances. Under memory-free conditions, these S-R instances must be assumed available throughout the course of discovery.

\(^{25}\)Although it is always possible to construct a rule that works with a finite number of given instances, extrapolation to other instances can never be guaranteed. A rule which works on the first n S-R instances,
The process of discovery learning, then, may be viewed as a sequence of tasks (goal situations), each of which is to be solved in a way that is compatible with the (available) solutions of the previous tasks. By compatible is meant that the previous goal situations can also be solved by the solution rule.

In accordance with the foregoing distinction between simple performance and learning (problem solving), there are two general kinds of situation that may arise. The first is where the subject knows or is given a number of possible rules and is asked to find which ones (one) work(s) on all of the instances presented. The second case requires, in addition, that the subject generate new rules during the course of the discovery. In this case, he must not only solve each of the tasks presented but he must come up with a solution that works on each of the preceding tasks.

Selection from Available Rules. Consider a situation in which the subject is presented with a series of tasks and is asked to "discover" which of the four basic arithmetical operations (+, −, ×, ÷) will serve to solve all of them. Suppose further that a variant of the "anticipation" method is used in which a series of stimuli is presented, in turn, and the subject is asked to give the response that he thinks goes with each (stimulus). After each response, the subject is told the correct response. To make the discussion definite, consider the instances shown on the next page, where the to-be-discovered rule involves subtracting one digit in a stimulus triple from another (digit).

For purposes of this analysis, we do not require the subject to remember anything. All relevant rules and previously presented instances are displayed before him at all times. In addition, a distinction must be made between the goal conditions, that the experimenter uses to determine the pattern of feedback he provides, and the effective goal situation for the subject, which will necessarily vary as a function of the available

---

25(continued) for example, can always be made inapplicable on the (n+1)st. On the other hand, it is always possible to generalize a given rule so that it applies to any finite number of additional instances. We illustrate the basic idea with a simple example. Suppose that r is a rule that works on the first n instances, but not on the (n+1)st. Let the response to the ith stimulus, i = 1,2,...,n be denoted r(i). Then, the rule values r(i) + s(i)(i-1)(i-2)...(i-n) represent the desired responses r(1), r(2),...,r(n) to the first n stimuli, respectively, and r(n+1) + s(n+1) n! to the (n+1)st stimulus, where s is a suitably chosen rule.

26This procedure parallels that traditionally used by experimental psychologists to study paired-associate learning, with one major exception. Each stimulus is presented only once and the subject's job is to learn to anticipate the correct response each time he sees a new stimulus.
instances. The correct response on each trial depends only on the task-defining (experimenter's) goal condition $G_E$ and the particular stimulus in question. In the present case, suppose that $G_E$ is $R = d_1 - d_3$ where $R$ is the response number, $d_1$ is the first digit of the triple and $d_3$ the third.

The subject, of course, is not told that $G_E$ is the desired goal; doing so would be tantamount to telling him which rule to select. At the start of learning, before any feedback has been given, all the subject knows is that one of the four computational rules applies to each instance and that he is supposed to "guess" the number which goes with each stimulus triple.\(^{27}\)

Given these boundary conditions and the mechanism postulated in Section 2, "learning" might be expected to progress as follows. On the first trial, effective goal situation $<S_1, G_1>$ is characterized by the stimulus $(4, 8, 2)$ and the goal "Find a response $R_1$ such that $R_1 = d_{11} \odot d_{13}$ where $\odot = +, -, \times, \div$." Each of the four available rules (i.e., $+, -, \times, \div$) may satisfy $G_1$, so according to our simple performance hypothesis, the learner will use one of them.

Suppose the addition rule is selected. In this case, the subject will respond and the experimenter will say something like "No! The correct response is 2."

On trial two, the effective goal situation is $<S_2, G_2>$ where $S_2 = (3, 5, 2)$ and $G_2 = \text{Find response } R_2$, such that $R_2 = d_{21} \odot d_{23}$ and $d_{11} \odot d_{13} = 2$ (where $d_{21}$ and $d_{23}$ refer to digits in the $i^{th}$ stimulus, and $\odot$ to a fixed computation rule). Notice in this case that the addition rule cannot be used as this would be inconsistent with part of the goal criterion ($d_{11} + d_{13} = 2$). There is no way to add digits in $S_1 = (4, 8, 2)$ and get $R_1 = 2$. Multiplication cannot be used for the same reason. The only "available"

\(^{27}\)Strictly speaking, there are more than four computational rules involved. Each possible pair of positions in the triples (e.g., first and third) defines a different rule. For example, one could introduce $(3 \cdot 2 =)$ 6 variations of each arithmetical operation, one for each possible pair of relevant attributes (positions). Alternatively, the available rule set might consist of encoding rules (for "reading" the stimuli) and the four computation rules, together with appropriate higher order abilities for combining these rules. This latter alternative, of course, would require problem solving on the part of the learner.
rules which may satisfy \((S_2, G_2)\) are subtraction and division.

Suppose, on trial two, that the subject selects the subtraction rule. In this case, the subject will give the response 1 and the experimenter will say "Yes. The correct response is one."

The same ideas apply on trial three, only this time the subject must take two previous instances into account. At this point, the division rule is no longer a valid candidate; it is not compatible with the second instance. Therefore, the only "available" rule, which may satisfy the goal on trial three, is the subtraction rule.

The subtraction rule, of course, was assumed to provide the basis for the pattern of feedback given by the experimenter (i.e., \(G_e\)) and is therefore consistent with each instance.\(^{28}\)

Notice that, in general, it is possible to predict the specific responses a subject will give only after all of the remaining and available rules (e.g., subtraction) are compatible with the task-defining condition. Before then, such predictions are not possible, given only the three hypotheses proposed in Section 4. The problem is that these hypotheses provide no way of telling which of the available rules on any given trial a subject might select.\(^{29}\) We simply assume that he will select one of them and let it go at that.

Nonetheless, according to our assumptions, the responses the subject gives must be compatible with the effective goal on any given trial. It is therefore possible to eliminate certain initially feasible rules from consideration as more information becomes available to the subject. The class of available rules either remains the same or becomes smaller with each succeeding trial. In effect, a sequence of trials imposes a non-strict linear order on the power set\(^{30}\) of rules that is available (and relevant) at the start of learning. Notice, in particular, that a given set of instances may be incompatible in the sense that none of the initially available rules applies to all of them. In this case, the empty set lies at the apex of the ordering. In most experimentation, the instances are selected so as to eliminate all but essentially just one rule, the

\(^{28}\)In real life it is often the case that none of the rules known to a subject are consistent with the available instances. In order to succeed in this case, the subject would be forced to "learn" a new procedure which does apply.

\(^{29}\)The problem of rule selection is considered in the context of motivation in Chapter 8. Particular attention is given to the question of why subjects select the specific rules that they do select where they have more than one rule available for achieving a given goal.

\(^{30}\)The power set of a set is the set of all subsets of that set.
one the experimenter wants the subject to "learn" (i.e., learn to select).

This observation raises the general question of what kind of prediction can be expected, on the basis of the available hypotheses, where more than one available rule is compatible with the subject's goal on a given trial. In this case, it is simply not possible to say in deterministic fashion whether or not the subject's response will satisfy the task-defining condition. On the other hand, predictions are not probabilistic either as any number of responses can generally be eliminated completely as possibilities.

In this kind of situation, we must think in terms of nondeterministic predictions. Unlike deterministic predictions, in which particular outcomes are uniquely specified, and unlike probabilistic predictions, in which probability distributions (of outcomes) are specified, nondeterministic predictions involve identifying subsets of possible outcomes without stating any bias in favor of one or another of the elements in these subsets. In making nondeterministic predictions, the theorist can usually eliminate certain outcomes from consideration entirely. He cannot, however, say anything about which of the possible outcomes is more "likely" to occur.

Stochastic theories may be viewed as approximations of deterministic theories whereas nondeterministic theories are more aptly viewed as less structured versions of deterministic theories. The general rationale underlying stochastic theories is that there are always unknown and/or uncontrollable factors affecting experimental outcomes and that theories ought therefore to deal with averages. The basic variables and hypotheses typically are selected because they provide a general overall account of the data, not because they (necessarily) account for particular outcomes. In stochastic learning theories, for example, although the hypotheses are typically stated in terms of probabilities concerning individual behavior, their predictive value is limited almost entirely to group statistics (e.g., means). Such hypotheses rarely say anything of value about individual behavior in particular situations. In principle, this will happen only where the probabilities are 0 and 1.

Nondeterministic theories are of a different sort. The point is that when one fails to uniquely specify a particular outcome, it is because the theory simply does not deal with the behavior in question. It is not a question of group statistics but of incomplete structure (i.e., a lack of relevant theoretical assumptions). In certain cases, this may leave open the possibility of "enriching" the theory later on so as to make deter-
ministic predictions possible. The purpose of Chapters 8, 9 and 10, in fact, is to do just this. Chapter 8, for example, deals with the problem of rule selection (motivation) under memory-free conditions.

Not all discovery learning, of course, takes place exactly as in our illustration. One of the more common kinds of variation involves the way in which reinforcement is given. In much experimentation, for example, the subject is told simply whether he is right or wrong. Telling the subject that he is right provides exactly the same information as telling him the correct response, but telling the subject that he is wrong does not. Negative reinforcement indicates only that the rule used on the trial in question is inappropriate. Knowing the correct response, in addition, frequently also makes it possible to eliminate other available rules from consideration on future trials.

Levine's (1966) work on simple discrimination learning provides an interesting example of the right-wrong type of situation for which a precise theoretical model is available. Translating his results into present terminology, Levine has shown that when there is a small number of rules available to the subject at the beginning of an experiment, telling him that he is incorrect leads him to reject the rule he used on that trial and to resample from the original class (of available rules) without replacement. Telling the subject that he is correct, however, effectively eliminates all rules in the original class which could not account for this response. Resampling on future trials, then, may involve a much smaller class (of rules).

In this case of simple discrimination learning, both Levine's model and the present form of analysis yield essentially the same kind of predictions. On "incorrect" trials subjects reject only the rule they are using, whereas on "correct" trials they may be able to eliminate entire classes of inappropriate rules. Unlike Levine's model, however, the present form of analysis is perfectly general and may be applied to any clearly defined discovery situation in which the subject has at least one appropriate rule available from the start.

It is also instructive to consider how associative learning might be viewed in these terms. The important point is that under the memory-free conditions proposed, subjects would be expected to respond correctly after the correct response is made available on the first trial. This follows axiomatically since a record of all of the responses given as feedback is assumed to be directly available to the subject throughout learning. For obvious reasons, no one would propose running a paired-associate experiment
under these conditions. Justification for such research would necessarily have to depend on whatever insights it might provide about perceptual or memory processes.

**Discovery of New Rules.** The discussion above has been limited to the case where the subject has all of the necessary rules available from the start. Explanation was primarily a matter of specifying how effective goal situations change over trials. The only behavior mechanism needed was that proposed in Section 2 to explain simple performance.

It is also possible to conceive of more complex forms of discovery in which the task on each trial is a problem to be solved. The overall task is made more complicated, however, by requiring a common solution procedure for all such problems. It is worth noting that this is the only type of discovery situation which truly qualifies as learning.

This more general type of discovery is not uncommon in the ordinary classroom situation where the teacher typically presents the desired goal directly and leaves it to each student to check his own responses. (This is feasible, of course, only where goals can be defined independently of rules adequate for achieving them.)

Two illustrative tasks of this type are shown below.

<table>
<thead>
<tr>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stimuli</strong></td>
<td><strong>Responses</strong></td>
</tr>
<tr>
<td>$x^2+5x+6=0$</td>
<td>$-2, -3$</td>
</tr>
<tr>
<td>$x^2+6x+9=0$</td>
<td>$-3, -3$</td>
</tr>
<tr>
<td>$x^2+2x-8=0$</td>
<td>$-4, 2$</td>
</tr>
<tr>
<td>...........</td>
<td>...........</td>
</tr>
<tr>
<td>...........</td>
<td>...........</td>
</tr>
</tbody>
</table>

The goal in Example 1 is to find the (integer) roots of each quadratic equation. Assuming that the subject can substitute correctly to check possible solutions, he presumably tries a variety of alternative rules until he finds one that works on all problems in a reasonably efficient way. (Clearly, one could try to check pairs of integers, in turn, but this would be extremely inefficient and, in general, unrewarding.)

This task has played a prominent role in some of the newer mathematics curricula, and has been presented successfully to many classes of fourth, fifth, and sixth-grade children. Oddly enough, the children tend to discover the sum-product rules for finding roots, which were a standard part of now largely defunct college courses in the theory of equations. They
almost never discovered any version of the quadratic formula of high school algebra. Presumably, this "result" can be attributed to the nature of the relevant rules typically available to the upper elementary school child.

In Example 2, the goal might be to find the sum as quickly as possible. The "as quickly as possible" is added to insure that the learner is encouraged not to use sequential addition, which would be extremely laborious on many problems of this type. Under these conditions, many subjects will discover that the sums can be determined by squaring the number of terms (in the number series).

Clinical experience, as well as certain experiments (e.g., Scandura & Voorhies, 1971) indicate that subjects typically go about solving such tasks in a systematic fashion. Random behavior apparently is a last resort where the subject has no other means available to perform a task—as is frequently the case with tasks used in experimental studies. This suggests that subjects may use higher order rules for constructing the lower order rules that they actually try.

Although we shall not attempt a detailed analysis here of either of these discovery situations, it should be noted that the problems involved are not of a conceptual nature. It would be a small task to generalize the analyses given above to include the case where the subject has to derive a new rule on each trial. The assumptions of Sections 2 and 4 allow nondeterministic predictions as before.\[31\]

6. EXPOSITORY LEARNING

Learning by exposition involves going from description to meaning (e.g., from a metalanguage to the corresponding object language). In other words, in order to learn by exposition, subjects must "interpret" some given discourse. This discourse may involve a natural language, mathematical symbolism, icons, or some other form of representation.

In view of the present theoretical commitment to rules generally, and the proposed learning and behavior mechanisms in particular, it is

\[31\]In realistic situations of this sort, the central problem in making predictions would be that of coming up with a set of rules which provides an adequate basis against which to measure what given subjects know when they enter the situation.

One could avoid such problems in experimentation, of course, by constructing artificial discovery situations in which the rules identified and made available to the subject can only be combined in certain predetermined ways to solve the task. The tasks used in the experiments reported in Section 4.2 could easily be adapted for this purpose.
natural to inquire into the nature of the process (of expository learning). If the needed constructs are not rule-like, for example, then the present formulation would suffer a serious blow. A positive answer, on the other hand, would lend considerable additional support for adopting the rule as the basic unit of behavioral analysis.

The general question of reference, of course, is extremely complex. All that can be done here is to sketch the manner in which rules and the proposed mechanisms may be involved. In Section 6.1, we attempt to clarify the nature of the interpreting task itself and to point out certain relationships to other ideas discussed in this book. Section 6.2 considers in somewhat more detail the nature of the interpretative mechanisms associated with different kinds of minimal signs (e.g., morphemes). Particular attention is given to differences in the way icons and symbols denote. In Sections 6.3-6.5, we show how human beings may "figure out" the meanings of arbitrary statements that they have never seen before. Emphasis is put on the need for a hierarchical system of interpretative abilities of the sort discussed throughout this chapter. No new behavior mechanisms appear to be needed.

6.1 The Nature of Interpretation

Inquiry into the nature of interpretation within the present framework raises a number of questions. It is not immediately clear, for example, what kinds of responses are involved, nor how to account for individual differences in the meanings assigned to particular signs.

Recall that in Chapter 2 we spoke of two distinct senses in which (meaningful) stimulus signs may be viewed. (1) Signs may be interpreted in terms of what they represent. Thus, signs (e.g., "5," "5," "5," etc.) may be held equivalent if they have the same meaning. This view seems most appropriate in dealing with meaningful behavior; "meaningful" stimuli might even be defined as stimuli which have clear referents. (2) Signs may also be thought of as meaningless entities in their own right, with properties of their own. In this case, signs are held equivalent according to whether or not they have certain properties in common. Even signs like "XPZ" and "*o+^!", which have no well-defined referents, for example, might be taken as equivalent since each has three perceptually distinct parts.

The problem of reference is one of explicating the relationship between equivalence classes of signs, qua signs, and their meanings. In the present section we make a beginning by proposing answers to two impor-
tant prior questions: (1) What are meanings? (2) How can one tell that a meaning has been determined (by a subject)?

In accord with our discussion in Chapter 6, signs belong to syntax and meaning to semantics. There, it will be recalled, an important difference was noted between the mathematical definition of semantics and the characterization of semantic knowledge. Specifically, meaning (semantics) in standard set-theoretic formulations is defined in terms of classes of entities in an object language. Semantic knowledge, on the other hand, is defined in terms of finite rule sets. For example, the meaning of an algebraic system is defined in set-theoretic terms as the class of embodiments of the system, whereas knowledge of the system is a set of rules which relate these embodiments. Furthermore, whereas the set-theoretic formulation suggests that knowledge of a system be characterized directly in terms of isomorphisms between the various embodiments, we saw that this was infeasible due to the indeterminately large number of isomorphisms that would be required. Higher order rules were introduced to reduce this number but, even so, one could not reasonably expect a rule set to account for all isomorphisms. In this sense, the set-theoretic definition corresponds to asymptotic knowledge.

Translation of these ideas to simpler meanings is direct. The sign "5," for example, is defined mathematically to mean the class of all sets containing five elements. This definition corresponds directly to the set of all one-to-one transformation rules between sets of five elements. Since no one could possibly learn all, or even a substantial proportion of such rules directly, our task is to show how the introduction of higher order rules may effect a substantial reduction in the number of rules required.

For example, it is easy to conceive of a higher order rule that operates on specific one-to-one transformations between sets containing some finite number of elements (e.g., three) and generates a corresponding one-to-one transformation between sets, each containing a different number of elements of the same kinds as in the first transformation. To illustrate, the transformation \(\{0, 00, 000\} \rightarrow \{+, ++, +++\}\) between embodiments of the number three might be mapped into the corresponding transformation \(\{0, 00, 0000, 00000\} \rightarrow \{+, ++, ++++, ++++, ++++\}\) between embodiments of the number five. More generally, knowing transformations between embodiments

32Definition of semantics in terms of canonical entities, and (effectively) classes of allowable transformations on them, is closer to the latter view in that the emphasis is on operations.
Mechanisms of Learning and Performance

of one number, together with appropriate higher order rules, may provide a basis for generating transformations between embodiments of other numbers.

Several points are worth emphasis. First, the meaning attributed to a given sign may vary over individuals. For example, to one child the meaning of the sign "5" might correspond to the class of those sets which can be put into one-to-one correspondence with a hand full of fingers, while to another child, the meaning might simply be a particular physical configuration. In no case can "5" mean something that the subject has not already acquired.

Second, there is no reason to suspect that the mechanisms by which meanings are learned are somehow different from those discussed previously. In view of the large number of transformations involved, however, and similarities with the Piagetian notion of number conservation, it is not surprising that it takes a long time for children to learn the "true" meaning of number.

Third, a given higher order rule may belong to any number of rule sets, each of which characterizes a different meaning. In this sense meanings may overlap (i.e., contain the same higher order rules). Meaning units, therefore, are defined not so much by what they contain but by the rules which operate on and generate them. Thus, a rule set that is outputted by an interpreting rule (assuming for the moment that they exist) would constitute a "meaning" by definition. Similarly, rule sets, which are operated on by rules for generating descriptions, would also constitute meanings. (Given the formalization in Chapter 5, one might also expect decision making capabilities to play a central role in defining meanings.)

There seem to be two basic ways of finding out whether or not the intended meaning of a given sign has been determined. One way involves describing or paraphrasing the meaning of the given sign. Thus, for example, a person who knows a correct meaning of the sign "5" might indicate this by saying something like "It means the class of all sets with numerosity five." In this case, we are effectively assuming that the subject has some means at his disposal for describing the meanings he has (e.g., the ability to use English).

The second way to determine whether the subject "knows" the meaning...
of a sign is by testing him to see if he can perform in accordance with
the meaning. For example, a subject who understands the statement "Incre-
ment the larger number as many times as the smaller" should be able to
give the appropriate responses when presented with various pairs of numbers.
Thus, if presented with "(5,4)" we could expect the response "6,7,8,9."
With a simple sign like "5," about all we could reasonably expect would be
for the subject to in some way display (i.e., draw, select, etc.) sets with
the appropriate number of elements and/or to pair arbitrary sets contain-
ing five elements in one-to-one fashion. Furthermore, since meanings may
be characterized as rule sets, it would be possible, in principle at least,
to determine meanings precisely by extending the methods of Section 3 on
assessing behavior potential. If such assessment proves practicable, this
could lead to some interesting hypotheses concerning the behavior of young
children.

Generally speaking, then, the typical situation in which expository
learning takes place involves presenting the learner with a sign (or com-
plex of signs) and requiring him to "come up" with the corresponding
meaning. In particular, we may present the subject with a sign and either
ask him to paraphrase the idea in his own "words" or to perform some tasks
involving the intended meaning.

Requiring the subject to paraphrase has a major disadvantage. It is
possible to paraphrase many statements without having the foggiest notion
as to their meanings. Any statement of the form "If A, then B," for
example, may be rewritten without any reference to meaning in the equiva-
1 lent form "B whenever A." The competencies involved in that case are
strictly syntactic, but an observer would have a difficult time distin-
guishing a paraphrase obtained in this way from one based on meaning.

In the second type of situation, criteria can be specified directly
in terms of performance that knowing the meaning of a sign complex makes
possible. Thus, for example, we might present a subject with the state-
ment "Increment the number $a$ by one $b$ times" and then ask him to apply the

34Although quite different on the surface, this approach bears some
similarity to Osgood's (1953) S-R formulation, in which responses are also
viewed essentially as indicators that signs have certain referents. Os-
good's (cf. 1966, 403) formulation may be represented $[S] \rightarrow [R]$, where $[S]$ denotes a sign (e.g., "5"); $[R]$, the meaning of the sign (e.g.,
the class of sets with numerosity five); $[s]$, its stimulus properties; and
$[R]$, the observable indicator (e.g., "00000"). The present view, however,
is both more precise and more general. For example, with signs having
well-structured meanings, the indicators of meaning can be made highly
specific and unambiguous. This view is also more general in the sense
that it allows rules (and associations) as meanings as well as simple
responses.
rule to enough different values of the variables $a$ and $b$ to insure complete knowledge of the rule.

Although it may help answer certain questions, this discussion raises some others. For one thing, the decoding abilities which need to be assumed in paraphrasing a given meaning, say, are not nearly as elementary as those associated with most of the behaviors considered so far. In setting up a behavioral test of this sort, then, an experimenter must take special care to insure that the subject has whatever decoding skills may be required. One would not ask a five year old, for example, to describe the meaning of the number five in words. In this case, it would undoubtedly be better to ask him to perform on matching tasks and the like.

Before moving on, another comment seems in order. For the first time we have come across a situation where it appears that a good deal of the learning involved may be of the "associative" variety (e.g., as in attaching "5" to its meaning). (This is not true of expository learning in general, however, as we shall see in Section 6.3.) My own inclination is to view association formation as an all-or-none phenomenon (e.g., Rock, 1957; Estes, 1960) when the stimuli and responses are already available as integrated units (e.g., Scandura, 1965). Rephrased in present terminology, association formation results upon application of a higher order rule to previously independent elements.\textsuperscript{35}

\textsuperscript{35}There are situations which do not involve meaning, of course, where detailed analysis of decoding processes also may be required. For example, if there were any question about a given subject's ability to write the numeral "5," then the procedures used to construct the numeral would have to be brought directly into the analysis (of the behavior).

\textsuperscript{36}Unfortunately, most of the associative learning studied in traditional experiments does not involve well-integrated units, and sufficiently many associations are involved on each trial as to make the memory-free hypothesis untenable. To this extent our treatment is not complete. (It is always easier to avoid a difficult problem by saying that it is beyond the scope of the present discussion than it is to meet it head on.) Nonetheless, the mechanisms to be proposed in Chapter 10 deal with a number of issues which are closely related; and it is not inconceivable that an account of paired-associate learning, say, might possibly be derived in terms of the mechanisms outlined there. In learning lists of paired-associates, for example, the limited capacity of human beings to process information might possibly interact with the fact that most stimuli and responses are not well-integrated units to produce what appears to be incremental learning. We shall not attempt to derive such results ourselves but, rather, will try to stay close to those issues which bear most directly on the learning of mathematics and other structured materials. I do hope, however, that others will be encouraged to look more deeply for possible interrelationships between existing theories of verbal learning (cf. Underwood & Schultz, 1960 and Cofer, 1963) and the sort of theorizing being proposed here. (continued on next page)
6.2 Properties of Minimal Signs (Symbols and Icons)

In this section, we ignore individual differences and think of meanings in terms of classes of denoted entities.

Minimal signs may denote classes of states (relations) or classes of operators. For example, the sign "5" denotes the number five, which is a class of states (1-ary relations). The sign "greater than" also denotes a class of states, only this time the states are binary relations. That is, the meaning is a class of binary relations. Other signs refer to operators or "actions." For example, "count," "double," "add," and so on, all define operators and, hence, refer to classes of actions. In English, verbs generally play this role. Thus, for example, "go" refers to the class of all acts of going and "run" to the class of all acts of running.

The meaning of a minimal sign, however, is not necessarily fixed once and for all, but may depend on the context in which it appears. For example, the meaning of the term "sum" involves a state when embedded in a statement like "Find the sum of 7 and 5," whereas it involves action when embedded in "Sum 7 and 5." Meanings which depend on context are beyond the scope of the present discussion.

In this section we consider some of the relevant properties had by different kinds of minimal signs. Particular attention is given to the way icons and symbols refer since they both play a central role in the exposition of mathematics.

Symbols. Probably the single most important characteristic of minimal symbols (morphemes) is that they denote arbitrarily. The arbitrary nature of symbol reference has both limitations and advantages. Perhaps its most important limitation is that (minimal) symbol reference is nongeneralizable. For example, there is no common way in which the numerals "5" and "6" refer. The meaning of each symbol must be learned separately; knowing that "5" denotes the number of elements in \{0 0 0 0 0\} does not help in learning that "6" denotes the number of elements in \{0 0 0 0 0 0\}. Any other symbol would be an equally valid candidate.

---

36(cont'd.) Moves in this direction will be necessary if we are to develop truly integrated theories of learning and behavior which apply to complex learning as well as to rote learning of the sort which has been studied most thoroughly in the experimental laboratory.

37The term "minimal sign," as used here, corresponds to what linguists call morphemes, minimal strings of symbols with meaning (e.g., words). We use the more neutral term "minimal sign" because we wish to consider icons as well as symbols.

38Actually, "5" and "6" originally may have been adopted from the icons "\(\overline{5}\)" and "\(\overline{6}\)," respectively, for which there is a general reference rule.
On the other hand, because symbols may be assigned arbitrary meanings, they can be used to represent highly abstract notions in a precise way. Thus, "five apples" refers to the class of all sets of five apples, whereas "5" refers to the class of all sets of five elements; but, there is no loss of precision associated with the increasing degree of abstraction. The symbol "N" (the set of natural numbers) refers unambiguously to a still higher order collection.\textsuperscript{39} Abstract relations may be denoted by symbols with equal ease. For example, the terms "taller-than," "greater-than," and "relationship-between" refer to progressively more abstract relations with equal precision.

Icons. In contrast to symbols, icons\textsuperscript{40} have properties in common with the entities they denote; they denote in a nonarbitrary way. This characteristic has important implications. In the first place, some relations seem easier to denote using icons than others. Proximity and relative size can be handled quite easily, but, as an example, the relationship between parents and their children can only be dealt with indirectly. Insofar as mathematics is concerned, icons seem to be particularly well suited to representing geometric ideas where the relationships involved tend to vary continuously. Icons (other than "moving pictures") do not represent operations (actions) directly, however, and require "arrows," or other icon-like signs, which are assigned meanings as operators.

On the other hand, because it is not entirely arbitrary, icon reference may involve nondegenerate rules. The icons "/", "//," "///," "////," etc., for example, can all be mapped onto their meanings by a common rule. This is possible because each icon can be put into one-to-one correspondence with the elements of the sets in the corresponding denotative class of sets. (That is, each set in the denotative class contains the corresponding number of elements.) For a second example, it is sufficient to note that particular properties of relief maps correspond to features of the terrain they represent. These corresponding features provide a suffi-

\textsuperscript{39}One of the major reasons why mathematics makes an ideal subject matter for study is that the denotations of mathematical symbols are typically equivalence classes. This clearly is not always the case with ordinary language. It is my guess that the more closely the major ideas of a subject matter approach the ideal of being well-defined, the more useful will the present approach be in formulating behavioral research in that area.

\textsuperscript{40}Here, "icon" is used to refer to any still or moving picture-like representation. While still pictures may refer to things and certain kinds of relations, moving pictures are required to represent action.
cient basis for constructing general rules for interpretation.

This ability of icons to refer in a generalizable way, however, is bought at a price. Because they must be referent-like, icons retain progressively more irrelevant information when used to represent increasingly abstract ideas. Note: It is easy to find an icon that can be used to represent a particular finite (arithmetic) sequence of numbers in which the successive numbers increase by a common amount. The sequence 1, 3, 5, 7, for example, can be represented by the icon

```
+---+
|   |
+---+
|   |
+---+
|   |
+---+
|   |
+---+
```

However, without the introduction of symbols of one sort or another, icons are not capable of representing arithmetic sequences in general. In this case, the icon would have to indicate, on the one hand, that there is a common difference between successive terms and, on the other hand, that both the number of terms and the relative size of the first term and the (common) difference between terms is irrelevant. Abstracting from the icon above, we observe that

```
+---+
|   |
+---+
|   |
+---+
|   |
+---+
|   |
+---+
|   |
+---+
|   |
```

would provide an adequate representation if it did not specify a relative size between the first jump and the successive jumps as well as a specific number of terms (i.e., 4). This information is irrelevant and, worse, misleading.

It should also be apparent that signs evident in the "real world" are like icons, only more so. Rather than being two-dimensional, these signs have three dimensions. Because of this, the signs and their referents will, generally speaking, have even more things in common. 41

---

41Still, it should be emphasized that "real world" signs need not refer to identity. To the contrary such signs almost invariably (cont'd.)
6.3 Characterization of Knowledge Underlying Interpretation

The ability to interpret minimal signs accounts for only a very small part of expository learning. If that is all humans could do, they could never even learn, for example, the meanings of more than a finite number of different Arabic numerals (e.g., "35", "153", etc.). Young children are undoubtedly limited in this way on first exposure to number, but it is equally clear that most people eventually do learn how to interpret arbitrarily large numerals. Furthermore, there are (nondegenerate) rules by which this may be accomplished.

It is possible to construct a rule for interpreting numerals of arbitrary size, but we can make essentially the same point more simply by just considering numerals with no more than two digits. In this case, the following rule will work

(1) Give meaning to the units-digit (i.e., the first digit on the right); then give meaning to the tens-digit; next, 'multiply' the meaning of the tens-digit by ten; finally, combine the meaning of the units-digit with the transformed meaning of the tens digit.

In order to interpret this rule properly, note the following: (a) Knowing the meanings of the digits 0 through 9 is basic to using the rule. (b) "Multiply by ten" may be interpreted to mean "Replace each element in each set in the denotative class of the tens-digit with ten elements of the same kind." For example, consider the numeral "35." In this case, we first give meaning to "5" as above. The same is then done for "3." In carrying out the next step, we refer to sets in the second meaning class (i.e., of "3"). Given the set {///}, for example, we construct the set [\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\33\7\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\34\35\36\37\38\39

Given that nondegenerate rules exist in the case of symbols as well as icons, we still face two kinds of problem. First, there is the general problem of how to account for arbitrarily large classes of interpreting behaviors involving a variety of complex signs. The fact that human beings acquire the necessary competencies undoubtedly stands as one of our greatest achievements; but the nature of these competencies has never been fully explicated, and the problem remains a major and continuing puzzle. The second problem is one of explaining how interpretation rules, like (1)

41(cont'd.) refer to broad classes. Thus, young children let blocks refer to automobiles, buildings, boxes, and so on. Even "John Smith," at a given instant in time, does not refer to identity--but, typically, to John Smith irrespective of when.
above, are learned. In both cases, the central question is whether or not the ideas proposed previously are adequate to the task.

In this section, we consider the former question of how to account for classes of interpreting behaviors. The next section deals with the problem of learning.

One alternative might be to characterize underlying competence in terms of a finite set of rules, each of which is presumed to act independently of the others. Although the simplicity of this approach has a certain appeal, it can be rejected out of hand for many of the same reasons discussed in Chapters 4, 5 and 6. With any inclusive class of interpreting behaviors, discrete characterization would almost certainly ignore many relationships between different interpretation rules. Any characterization of the ability to read Spanish, for example, is almost certain to have much in common with corresponding characterizations of the ability, say, to read Italian and other Romance Languages.

Although I know of no serious attempt to identify such relationships specifically, their existence seems undeniable. An adequate account of the ability to read in two or more Romance Languages would almost surely include rules relating rules needed to read in the various different languages, to say nothing of the probable need for rules relating (classes of) rules within languages.

These comments apply equally as well to "reading" mathematics. The ability to interpret different forms of mathematical description undoubtedly involves higher order rules, higher order rules which relate different notational systems for describing the same mathematical ideas, or parts of the same notational system in describing different ideas.

In order to account for such relationships, of course, one could simply add rules to a given discrete characterization as new relationships are uncovered. But, as argued in Chapters 4 and 6, any characterization constructed in this way would have an arbitrary post hoc flavor and not be particularly enlightening.

Just accepting the principle that rules may operate on rules, of course, will not solve the problem.\footnote{Nonetheless, the higher order approach has important advantages because it does not require that each possible kind of rule be taken explicitly into account.} It is unlikely that anyone will soon succeed in coming up with a truly adequate account of any nontrivial class of interpreting behaviors. As important as such analyses might be, they are beyond the scope of this treatment. We shall adopt instead the far
more modest goal of choosing some small and reasonably well prescribed classes of interpreting behaviors and, then, showing how such behaviors may be accounted for in terms of rule sets in which rules may operate on (classes of) other rules.

We first show how the interpreting behaviors associated with two-digit numerals may be accounted for by a rule set that does not explicitly contain Rule (1) above. Instead, the rule set includes a higher order rule by which minimal interpretation rules may be combined to form new interpretation rules. In this case, the higher order rule takes the form

\[ n \rightarrow n', n' \rightarrow m' \Rightarrow n'n' \rightarrow 10m' + m \]

where \( n \) and \( n' \) are single digits and \( m \) and \( m' \) are their respective numerical meanings (i.e., classes of sets). In this case, \( 10m' + m \) refers to the class of all set unions, where one set is in the meaning class of \( m \) and the other set has 10 times the numerosity of one in the meaning class of \( m' \). In short, \( 10m' + m \) refers to the number represented by the numeral \( n'n' \).

Notice that the inputs of this higher order rule are pairs of associations (e.g., "5" \( \rightarrow \) 5, "4" \( \rightarrow \) 4) and that the outputs are also associations (e.g., "45" \( \rightarrow \) 40 + 5). In effect, although the rule set accounts for the same behaviors as rule (1), rule (1) itself cannot be derived from the rule set. To accomplish this, something more is needed. One possibility would be to add a generalization rule of the sort discussed in Chapter 5 to the rule set. Application of such a rule to output associations (e.g., "45" \( \rightarrow \) 40 + 5) would have the effect of replacing constants with variables and would yield the general rule \( n'n' \rightarrow 10m' + m \).

Assume now that we have a still more general rule that interprets arbitrarily large numerals. Furthermore, suppose that we want to account for the interpreting behaviors associated with numerals in arbitrary number bases. Is it necessary to add separate rules for each base? This would clearly be one possibility, but it is not necessary. Instead, we could add a higher order rule which, given the base ten rule and any specific base (e.g., five), generates a corresponding rule for interpreting numerals in the new base. Basically, what this higher order rule does is replace 10 with the given base throughout the base ten rule. 43

Finally, we consider the problem of how to account for interpreting

behaviors associated with compound mathematical expressions involving parentheses. Suppose, for example, that we want to construct a rule set that accounts for the interpreting behaviors associated with such expressions as

\[
\begin{align*}
(2) \quad & \Sigma x \left( s \cdot t \right) + \left( (([6.2] \div [7.8]) [2.4]) + [2.7]) \right) [8.3]; (3\cdot4)+5 \\
& x=1 \quad y=1 \quad z=1
\end{align*}
\]

and a wide variety of others. One way to do this would be to simply introduce a rule for interpreting each kind of expression. This would be perfectly alright as long as the number of kinds of expression is reasonably small. Where this is not the case, however, one might attempt to capitalize on communalities among the expressions. With this in mind, another possible rule set might include a higher order rule for using parentheses—for working from the inside out, together with rules for interpreting the simple sums, differences, products, and/or quotients in the various expressions.

In accounting for interpretation behavior, then, the parenthesis rule may operate on the simple interpretation rules and output new rules for interpreting compound expressions.

For example, represent two simple interpretation rules as

\[
\begin{align*}
s \quad t \quad & \rightarrow \text{Msum} \\
\Sigma y + \Sigma z \quad & \rightarrow \text{Mproduct} \\
\end{align*}
\]

where \(\text{Msum}\) and \(\text{Mproduct}\) denote the respective meanings. The parenthesis rule applies to expressions and simple interpretation rules, and may be described

(3) Construct an interpretation rule in which those simple interpretation rules that apply to expressions in innermost parentheses are applied first; add other rules in the order required in working from the innermost parentheses out.\(^{44}\)

In order to interpret the first expression of (2), rule (3) is applied to this expression and the simple interpretation rules above to generate a rule which interprets \((\Sigma y + \Sigma z)\) first, and then the multiplication. Notice that we have here an example of a higher order rule that operates in part on simple stimuli.

In effect, given a rule set which includes rule (3) and rules for interpreting simple expressions of the above sort, involving sums, differences, products, and quotients, it would be possible to account for the

\(^{44}\)This rule ignores a number of details which are inessential for our purposes.
ability to interpret any given expression whatever that can be constructed from the given simple expressions by use of parentheses. Adding rules for interpreting new kinds of simple expressions, involving the arithmetical operations, would further increase the power of the characterizing rule set.

We could even go one step further and replace the simple interpretation rules (for interpreting expressions involving one binary operation) with minimal rules for interpreting the minimal symbols involved (e.g., $3$, $\sum_{x=1}^{2} x$, [7.3], etc.). In this case, the characterizing rule set would consist of these minimal rules, the parenthesis rule, and other higher order rules which operate on the corresponding minimal rules to generate simple interpretation rules. A higher order rule of the last-mentioned type for addition may be denoted

$$n_1 \rightarrow m_1, n_2 \rightarrow m_2 \Rightarrow n_1 + n_2 \rightarrow m_1 + m_2$$

where the $n_1$ and $m_1$ are minimal symbols and their meanings, respectively.

The potential power of this type of characterization would be even greater than before, in the sense that the power of the rule set may be increased by simply adding minimal rules for interpreting new minimal signs. There would be no need to add a new simple interpretation rule, involving addition for example, every time one wanted to introduce a new type of symbol. Even more important, the introduction of higher order rules, involving new binary operations, would automatically make it possible to interpret arbitrary expressions involving the new operations (and/or other available binary operations).

6.4 Mechanisms of Learning and Performance in Expository Learning

As one might suspect from the preceding analyses, the mechanisms described in Sections 2 and 4 provide an adequate basis for explaining how interpretation rules are learned and how they are put to use. Suppose, for example, that a subject knows the various meanings of the digits 0, 1, 2, ..., 9 and that he also knows a higher order place value rule for combining the digit rules. (We suppose further that the subject has the decoding abilities required to demonstrate that he knows any given meaning, but that he has not learned a general rule for interpreting numerals.) The task here is to show how the subject is able to put this knowledge to use to interpret multi-digit numerals. To make things definite, let the stimulus be "52."
Since the subject, presumably a young child just learning place value, is assumed not to have a procedure immediately available which applies in this situation, he must "figure out" what "52" means. According to our analysis, he does this as follows. First, control shifts to the higher order goal of deriving a procedure $p$ such that "52" is contained in the domain of $p$ and the range of $p$ contains numbers (i.e., meanings). By hypothesis, the subject has available the place value rule as well as the specific digit-interpreting rules. He therefore applies the place value rule to the digit rules "5" $\rightarrow$ 5 and "2" $\rightarrow$ 2 to generate the new rule denoted "52" $\rightarrow$ $10\cdot5 + 2$. This rule satisfies the higher order goal. Hence, control reverts to the original goal and the subject applies the newly derived rule. So goes the analysis.

In working with an expression like

$$\sum_{x=1}^{u} \left( \sum_{y=1}^{v} \sum_{z=1}^{w} \right) \frac{r}{s} t$$

we need to be more careful. Here, the parenthesis rule does not apply to minimal rules like $\sum_{w=1}^{u} m$ but rather to rules like

$$r \cdot s \rightarrow M\text{sum} \quad \text{and} \quad r \cdot s \rightarrow M\text{product}.$$  

In this case, we either can assume that rules of this latter sort have already been mastered or we can add another level to the analysis. In the latter case, we let $A$, $B$, and $C$, respectively, correspond to the three expressions of the form $\sum_{w=1}^{u} w$ and assume the following interpretative abilities:

(a) $A \rightarrow m_{a}$, $B \rightarrow m_{b}$, $C \rightarrow m_{c}$ (denoted $R_{1}$, $R_{2}$, $R_{3}$ respectively)

(b) $R_{i}, R_{j} \Rightarrow R_{+}$, $R_{i}, R_{j} \Rightarrow R_{x}$ (denoted Rule Sum, Rule Product)

(c) Parenthesis Rule

where $m_{a}, m_{b}, m_{c}$ refer to meanings, $R_{+}$ is a rule for assigning meaning to simple sums and $R_{x}$, to simple products.

The analysis, then, would go roughly as follows. The subject does not have a procedure available for interpreting the given expression

$$\sum_{x=1}^{u} \left( \sum_{y=1}^{v} \sum_{z=1}^{w} \right) \frac{r}{s} t$$

so control shifts to the next higher level. But, the subject still does not have a higher order rule available for deriving a procedure of the required type so control shifts to the still higher order goal of deriving
a higher order procedure, for deriving a procedure that will work. At this level, the Parenthesis Rule applies to rules (b) and the expression giving the rule:

Do Rule Sum and then Rule Product.

Application of this rule, in turn, to \( R_1, R_2, \) and \( R_3 \) gives the required rule:

Do \( R_2 \) and \( R_3 \), "add" the meanings, then do \( R_1 \) and "multiply."

Finally, the required rule is applied to the given expression and the meaning is determined.

The above analyses show how our theory provides a basis for prediction in expository learning wherever the relevant entering capabilities of a subject are known. It also provides a basis for explanation, even where the available and relevant rules are not known directly. This is feasible because it is frequently possible to identify most, if not all, of the alternative rules that individuals might use to interpret given kinds of statements. Furthermore, these alternative rules constitute a competence theory of sorts against which individual knowledge may be measured by testing. Once the rules actually available to a subject have been identified, of course, we are right back where we started and can make explicit predictions concerning the subject's interpreting behavior.

Fairly direct experimental support is available for this type of analysis.\(^{45}\) In a recent study, Scandura (1967b; also see Volume II, Chapter 2) was able to show that where an underlying "grammar" has been mastered, knowing the meaning of particular minimal symbols (e.g., \( \frac{u}{w} \)) was a necessary and also essentially a sufficient condition for applying a compound rule whose statement involved these particular symbols. What was done, in effect, was to determine whether the subjects had acquired the intended meaning of (new) compound statements by seeing how well they could apply the underlying rule (See Section 6.1.) As in our analyses, the grammar involved parentheses. The originally naive subjects were trained with neutral materials (e.g., \( 3(5 + 4)(3 + 2) \)) until they could reliably work with parentheses. Then, half of the subjects were told the meaning of minimal signs like \( [x] \), the largest integer in \( x \). Before going on, each subject was required to demonstrate that he knew the correct meaning of each kind of minimal sign by responding appropriately with respect to a class of instances like

\^\(^{45}\)Unfortunately, at the time the study was conducted, my analysis of the interpretation problem had not yet reached the present (and still preliminary) state.
Under these conditions, once a subject had committed to memory a statement like

\[([(x) + (y)] \div [(z)]\]

he could almost invariably apply the indicated rule. That is, these subjects could interpret the statements and thereby solve problems like

\[([(6.7) + (2.4)] \div (3.1)] = [(6 + 2) \div 3] = [2.67] = 2\]

Those subjects who were not given the meaning of the minimal signs were uniformly unable to respond appropriately.

Although it constitutes but a small beginning, this experiment demonstrates the feasibility of studying interpretation phenomena in the laboratory. In conducting experiments of this kind, however, it must once again be cautioned that the analysis does not take memory limitations into account. Experiments must be designed accordingly.\(^{46}\)

6.5 On Ambiguity

Along with inference rules, rules for interpreting statements seem to occupy a special position in characterizing knowledge. Inference rules, recall, are universally applicable in the sense that they apply in all situations. As with other rules, however, any number of inference rules may apply in any given situation. Whether a rule is appropriate or not depends on the specific goal in question.

In expository learning, on the other hand, most statements can be interpreted unambiguously. Directing a subject to interpret a statement is usually tantamount to specifying a single rule (rule set) which gives the meaning. In fact, the reason that the English language is as useful as it is is that essential ambiguities are relatively infrequent. The same thing is true of other natural languages and with mathematical

\(^{46}\)These ideas could have important implications for the ability-treatment interaction question that has been current in educational research circles during the past few years. Roughly speaking, the basic idea is to match students to particular kinds of teaching (e.g., verbal, spatial, etc.) according to their preferred form of interpretation. My own guess is that if the idea is to bear real fruit, it is going to require a deeper theoretical base than those that have been proposed to date. In particular, I feel that we may need to move away from standardized measuring instruments and toward instruments designed specifically to get at basic interpretation abilities. Hopefully, these abilities will be smaller in number than might be supposed in view of the higher order capabilities likely to be involved; but, in any case, they will have to be identified before we can expect much progress, and it is clear that this will not be a trivial task.
symbolism.

This does not mean that individual phrases, or sentences, or even paragraphs, may not be ambiguous. The standard sentence "Flying planes may be dangerous," for example, may be interpreted in either of two ways:

(1) (Flying planes) may be dangerous.
(2) (Flying) planes may be dangerous.

Contrast this sentence with "Flying planes are dangerous," which has an unambiguous meaning corresponding to (1). Nonetheless, essentially all statements can be made unambiguous by embedding them in some larger context (which may be verbal or nonverbal).

One way of explaining the relative lack of ambiguity in most common languages is to suppose that either by design, or more likely evolution, only one interpretation rule applies to most statements. Where two or more interpretation rules apply, statements will be ambiguous.

7. SUMMARY AND CONCLUDING REMARKS

The approach to theory development reflected in this chapter differs substantially from that which has been current in psychology throughout its recent history. First of all, the theory is deterministic, rather than stochastic, which already breaks sharply with tradition. Second, the theory proposed is about individuals and individual behavior. It is a theory in which the subject himself, and more particularly, his relevant entering capabilities become the major parameters to be specified. Third, the theory rests fundamentally on theories of competence. The adequacy of explanation and particularly prediction depends on the extent to which the underlying competence theory in question provides an adequate basis for measuring human knowledge.

Equally important, the theory is a partial theory. That is, it is a theory which, although applicable in a wide variety of situations, deals only with certain kinds of behavior—in particular, the theory yields useful predictions only where the subject's goal is known and the role of memory is minimal. This is in contrast to miniature theories of the sort that have been current in academic psychology (particularly, mathematical psychology). In the latter case, the aim has been to devise theories which deal in a very thorough manner with learning and performance on a narrowly defined class of tasks.

The partial theory approach led us quite naturally into a discussion of nondeterministic theorizing as a basis for dealing with situations in which the theory was not explicit about what behavior to expect. The most
notable example came in our discussion of discovery learning where we were unable to specify precisely which responses would be given on pre-criterion trials.

In the next three chapters, we shall see how more structure can be added to our partial theory by introducing additional assumptions. These assumptions, in turn, make possible deterministic predictions in situations which previously could only be handled in nondeterministic fashion.

\[47\] From our discussion in Chapter 2, it should be clear that what are considered to be the effective responses in a given situation depend on the goal in question. Thus, deterministic predictions are possible in discovery learning with respect to the effective goal on each trial but not with respect to the task-defining criterion.
In concluding his well-received book on cognitive psychology, Neisser (1967, 304) says, "If what the subject will remember depends in large part on what he is trying to accomplish, ..., do not predictions become impossible and explanations post hoc?" In his closing paragraphs (p. 305), he goes on, "The simplifications introduced by confining the subject to a single motive ... can be justified only if motivation and cognition are genuinely distinct ... It is no accident that the cognitive approach gives us no way to know what the subject will think of next. We cannot possibly know this, unless we have a detailed understanding of what he is trying to do, and why. For this reason, a really satisfactory theory of the higher mental processes can only come to be when we also have theories of motivation, personality, and social interaction. The study of cognition is only one fraction of psychology, and it cannot stand alone."

As we saw in the previous chapter, this is something of an overstatement. A cognitive theory of learning and performance may lend itself to prediction and it may have important practical implications. On the other hand, Neisser is certainly correct in emphasizing that motivation must be dealt with. The problem is explored in this chapter. In Section 1, we look more closely at the nature of motivation itself as it relates to structural learning. Specifically, how does motivation relate to our earlier discussion and just what do explanation, prediction, and control involve? Section 2 is concerned with some of the kinds of data that might be used to determine what a subject is trying to do. In Sections 3 and 4, we consider some of the kinds of hypotheses that might be proposed to account for motivation. The emphasis is on prediction.

As with Chapter 7, the present treatment is not definitive but serves primarily as a general introduction to the problem. The formalization
in Chapter 9 pulls together and synthesizes the essentials of this and the preceding chapter, thereby providing a firmer foundation upon which more serious empirical and theoretical work may build.

1. NATURE OF MOTIVATION THEORY IN STRUCTURAL LEARNING

What does theorizing about motivation involve and how does this relate to our earlier discussion? In view of Neisser's remarks, we might be tempted to define the task of motivation theory as one of predicting which goal subjects will adopt in given situations and let it go at that. This would not be sufficient, however, for that would not tell us where such goals come from in the first place, nor how they relate to the situation at hand.

In almost every situation an observer has some idea of what a given subject is trying to accomplish. He may not know, for example, what sort of building an architect will design, but he can be quite sure that it will be a building. Similarly, he can usually be fairly certain that the next move made by a chess master will be a good one, although he may not know what the specific move will be. He can also be reasonably confident that, when faced with a simple theorem, a competent mathematician will come up with a valid proof; but generally speaking, he will not know what kind of proof it will be. Analogous statements abound.

The motivation theorist's task is to say something additional about what a subject will actually do in any given situation, whether this involves explaining why the architect designed the building he did, why the chess master made his particular move, or why the mathematician used an indirect proof, or the child, a short cut in addition. More particularly, the key question for motivation theory is to explain why the subject took, or will take, the path he did. In retrospect, it appears that we have already proposed an answer to this question in the special case where the subject has no rule available for achieving the initial goal.

The problem comes where the subject has more than one learned rule available for achieving the initial goal. It was assumed in this case that the subject would use one of the available rules but nothing was said about which one.

I contend that the answer to this question of "which one" lies at the base of what we normally think of as motivation, especially as it is realized in complex human learning involving structured subject matters.¹ To see why I believe this, observe that knowing which of several available

¹A similar suggestion was made earlier by Taylor (1960).
and relevant rules a subject will select makes it possible to say more about his specific responses than that they will satisfy the given goal. Although the responses generated by each such rule may be functionally equivalent insofar as the initial goal is concerned, they generally are not in other ways. There are many different ways of constructing buildings, for example, which are equally as good insofar as living space is concerned but not, say, in terms of resistance to fire (e.g., consider brick and wood-frame construction).

In effect, the responses generated by each relevant rule, available to a learner in a given goal situation, generally satisfy conditions that are over and above those associated with the initial goal situation. These additional conditions, together with the criteria associated with the initial goal situation, define a refined goal situation. Given a goal situation \((S_0, G)\), we let \((S_1, G_1)\) represent the refined goal situation corresponding to available rule \(r_i\), \(i = 1, 2, \ldots, n\).\(^2\)

The upshot of all this is that specifying which of the available rules a subject will select is tantamount to specifying which of the corresponding refined goals he will adopt. Hence, given any initial goal and assuming that the subject has available a finite number of rules for achieving that goal, the task of the motivation theorist is to explain why the subject chose the rule he did, and, if possible, to specify this rule in advance.

If explanation is the theorist's sole task, then all he need do is to make sure that he can explain in terms of the theory why the subject's responses satisfy the refined goal. It is not essential that he know

\(^2\)Where the observer does not know all of the rules a given subject has available, there may be some responses which do not meet any of the identified refined goal situations. This possibility is easily taken into account by simply introducing a new category of nonstandard responses, consisting of the complement of the union of the refined goal situations that have been identified relative to the original goal.

\(^3\)These observations provide a basis for resolving a long-standing disagreement between a colleague and myself. I have held to the view that classes of rule-governed behavior should be defined as functions. Only by doing so could we use deterministic procedures and, thereby, hope to devise deterministic theories of behavior. He has argued that behavior is basically relational in nature and that defining rule-governed behavior as functions would make it impossible to treat many important kinds of behavior, like those involved in playing chess. In a sense, we are both right. I still feel more strongly than ever that rule-governed behavior should be defined in terms of functions. But, now, I also see more clearly than I did before why certain types of behavior appear to be relational in nature. It is simply due to a confusion between the goal criteria used by the observer to evaluate responses and what he knows about what the subject is trying to do before the subject acts.
beforehand which relevant rules the subject has available. Indeed, the task of Section 2 is to show (after the fact) how to determine which of several rules provides the best overall account of the obtained data. On the other hand, if the theorist is to have any chance of predicting what the subject will do next, he must have some idea of what relevant rules the subject knows. Without this knowledge, he could not possibly predict anything.

Even knowing what relevant rules a subject has available, however, is not sufficient. In the previous chapter, recall that in general it was only possible to make nondeterministic predictions concerning performance. The first hypothesis said simply that the subject would select and use one of the available rules, but nothing was said about which one. In this chapter we shall attempt to increase the power of our theory by imposing more structure on it--that is, by introducing additional hypotheses which make more refined predictions possible.

To fully appreciate the scope of what needs to be done, it may be instructive to consider two extremes under which an adequate motivation theory must hold. At one extreme, no initial goal may be specified at all so that any response would meet the (unspecified) conditions. Presumably, this is the type of open-ended goal situation we have in mind when we speak of autistic and other apparently non-goal-directed behaviors. Although not strictly obtainable in the experimental laboratory, this type of goal situation is perhaps best typified by free-association tasks in which the experimenter wishes to say or predict something about the kind of responses a subject will actually give.4

At the other extreme, there may be only one refined goal situation, namely the initial goal situation itself. In this case, the situation is essentially that of Chapter 7 where we restricted the subject to a single goal.

The vast majority of situations, of course, lie between these two extremes. Many theorems, for example, have several different kinds of proof; the experimenter may want to say something about the kinds of proof particular people might give. Similar questions may be asked about playing chess, designing highway systems, and adding numbers. There is clearly

---

4In most free-association studies, of course, it would be practically impossible to take account of all of the rules that a given subject might use. This need not cause difficulty in making predictions, however, since the experimenter only needs to specify those refined criteria which for one reason or another are of particular interest to him. This makes it possible to ignore certain rules and to lump together other rules which satisfy common refined criteria.
more than one way to go about each such task, and it would be invaluable to have a theory that would allow us to determine which way given subjects are most apt to choose.

In the next section, we discuss some alternative ways of determining, after the fact, which rule a subject has used.

2. IDENTIFYING MOTIVES (GOALS) AND RULES - EXPLANATION

2.1 Verbal Reports

Perhaps the simplest way to determine what rule a subject used in a given goal situation is to ask him. Because of its roots in early "structuralism," however, the use of verbal reports has been held in generally low repute by experimental psychologists. Among the reasons given for avoiding verbal reports are that (1) they are hard to classify and, thereby, are difficult to score objectively, and (2) a subject is often unable to describe his internal processes--indeed, he may not even know what they are or were. Verplank (1962) recently obtained direct evidence for these contentions when he showed that subjects may be conditioned to respond one way in a concept attainment task but to report quite different reasons for their responses. Nonetheless, verbal reports do provide an independent and potentially valuable source of data and should not be overlooked.

2.2 Refined Goal Criteria

As indicated above, a subject may have any finite number of different rules available which apply in a given goal situation. Each such rule, furthermore, has attached to it a particular profile of responses which satisfy, in addition to the given goal criterion, more refined criteria which relate specifically to the particular rules in question. For example, although there are many different procedures for finding sums of numbers, such procedures may vary greatly in terms of efficiency. Thus, it would almost certainly take less time to add the numbers 52 and 47 using the addition algorithm than by counting.

This observation suggests another way to determine what rule a subject has used. Given a goal situation, the experimenter qua competence theorist first identifies all of the procedures that are likely to be used in that situation. Theoretically speaking, the number of such procedures is denumerably large but generally speaking, the number of feasible procedures is quite small. (Furthermore, it is possible to allow for other possibilities by introducing a class of procedures called "none of these.")
The next step is to identify the refined goal criteria associated with each of these procedures. (The complement of all these criteria relative to the given goal criterion corresponds to the class "none of these.") Finally, the subject's actual responses are compared with the various refined goal criteria to see which one(s) they fit best.\footnote{Each refined goal criterion actually defines a class of procedures, namely those procedures which yield responses which satisfy it. Since only one of the procedures in each such class is used to identify the corresponding refined criterion in the first place, however, the others can safely be ignored.}

Perhaps the clearest application of this general technique is in a study by Suppes and Groen (1967), which involved making (what they called) latency predictions on a simple addition task. The basic procedure was to identify a number of counting algorithms for summing small numbers and, after making certain assumptions about the time required to carry out each (type of) step in the algorithms, seeing which algorithm best accounted for the behavior actually observed. It was implicitly assumed that a single algorithm was used on all items in the specified class and that one of the five algorithms they identified was actually used.

The problems they considered were of the form $m + n = \_\_\_\_$, where $m$, $n \geq 0$ and $m + n \leq 5$. Although they did not describe these algorithms in exactly the same way as described in Chapter 2, it is clear that they could have done so and we shall consider the algorithms in the simplified form that they did use. What Suppes and Groen did, in effect, was to assume that certain rules (e.g., encoding rules) could be safely ignored. All of their algorithms were based on counting and were described in terms of a counter on which two operations were possible: (1) setting the counter to a specified value and (2) incrementing the counter by one. In each case, addition was performed by incrementing the counter by one as many times as necessary.

The five counting algorithms were as follows.

1. Set the counter to zero. Then, add $m$ and $n$ by increments of one.
2. Set the counter to $m$ (the left-most number). Then, add $n$ (the right-most number) by increments of one.
3. Set the counter to $n$ (the right-most number). Then, add $m$ (the left-most number) by increments of one.
4. Set the counter to the minimum of $m$ and $n$. Then add the maximum by increments of one.
5. Set the counter to the maximum of $m$ and $n$. Then, add the minimum by increments of one.
At this point, Suppes and Groen switched to a stochastic mode. They let the time required to carry out each of the basic operations be represented by random variables. Thus, \( \tau \) was the random variable representing the time required to set the counter, and \( \beta \), the time to increment by one. Since the time \( \tau_i \) required to generate a given response to task \( i \) is a function of \( \alpha \) and \( \beta \), \( \tau_i \) is also a random variable (for all \( i \)). In this case, \( \tau_i = \alpha + \beta x_i \) where \( x_i \) represents the number of times the counter is incremented. Notice that only \( \tau_i \) and \( x_i \) vary over problems.

To simplify the analysis, Suppes and Groen worked exclusively with expected values (i.e., means) in generating predictions. By standard regression methods, they obtained least squares estimates of the predictors \( E(\alpha) \) and \( E(\beta) \), and derived estimates of the expected latencies \( E(\tau'_i) \) on the various addition problems. There were five sets of predictions in all. For example, using algorithm (5), the expected time required for adding \( 2 + 3 \) is the time \( E(\alpha) \) required to encode 3 (the maximum) plus \( x_i = 2 \) times the time \( E(\beta) \) required to increment by one.

Comparing these estimates with the obtained results of an experiment with first graders, they found that the expected latencies generated by algorithm (5) did the best job of accounting for the data. This does not mean that all, or even any, of the subjects used this algorithm on all of the problems, but only that it gave the best overall account of the data.

In his dissertation, Groen (1967) did a follow-up study in which he estimated separate latency parameters for individual subjects. In this case, he found that algorithm (5) again gave by far the best fit for about half of the subjects. The data of the other subjects was not well accounted for by any of the algorithms. Groen found, however, that by assuming extremely short latencies for certain simple additions, like \( 1 + 1 \) and \( 2 + 2 \), he was able to improve the degree of fit obtained for many of the subjects, including that of some whose data was not accounted for by algorithm (5). Although Groen did not interpret his results in this way, it would appear that at least some of the subjects tended to use different algorithms on different problems.

It does not take much imagination to see how this procedure might be generalized to other situations. The latency estimation procedure described above, for example, might be extended directly to other types of computational problems. More generally, the refined criteria must be tailored to meet the needs of particular kinds of situations. Thus, for example, in proving theorems, the form of the proofs might be taken into account as well as their logical validity. The method described in Chapter 7, Section 3.6 may also be adapted for this purpose.
2.3 Further Comments

In order to place this discussion in the perspective of the main development, several points should be emphasized. First, this method is useful primarily as a means of accounting for data after the fact, not predicting it beforehand. The parameters (e.g., $\alpha$ and $\beta$) used in making "predictions," for example, are generally estimated from the data itself. It would be possible to estimate parameters on one set of data and make predictions on another set (i.e., to cross-validate), but the outcomes of such attempts have only rarely been satisfactory.

Second, it is one thing to identify a rule which gives the best overall account of a class of behaviors, as was done by Suppes and Groen, and quite another to identify a subject's basis for responding on specific instances. It does not necessarily follow that a subject will use a single rule on all instances of a class of rule-governed behaviors. Where more than one alternative is available to the subject, he may use one rule on one problem and a different rule on another.\(^6\)

Third, neither the methods described nor the studies cited in this section say anything about how or why subjects select the rules they do. They tell us something about which rules subjects tend to prefer but not why they prefer them. What we need to do is to formulate hypotheses about why and how subjects select rules as they do and to test these hypotheses. This is the subject of the next two sections.

**MECHANISMS OF MOTIVATION**

For present purposes, the problem of motivation theory is formulated as follows: Given a goal situation $(S,G)$, and a finite set of relevant rules which may satisfy $(S,G)$, why does the subject select the rule that

\(^6\)Asking a subject how he worked each particular instance could provide the necessary information, of course, but this implicitly assumes things about what the subject can describe and, perhaps, a less subject-dependent procedure would be more desirable. Let me illustrate by outlining briefly one possible approach. (No special virtues are claimed for the approach; indeed, I feel that there may be far better alternatives available (cf. Section 3).)

As in Section 1, the first step is to identify the various procedures that might reasonably be used on particular instances. (The variety of rules may not be the same for all instances in a given rule-governed class. For example, $2 + 2 = 4$ is typically learned as a "direct association," whereas $5497 + 367$ almost certainly would not be.)

Next, the refined criteria imposed by the identified procedures must be determined. These criteria might be determined directly, as in distinguishing between specific forms of proof (continued on next page)
he does?

In so-called hypothesis-testing theories (e.g., Restle, 1962; Levine, 1966, 1969), it is generally assumed that subjects select from among available concept attributes (rules) in strictly random fashion. This is probably a reasonable assumption to make in the highly artificial situations which have been chosen for study. The problems amount to simply guessing what (attributes) the experimenter has in mind. Even in artificial tasks of this sort, however, subjects frequently have very definite preferences for particular rules (e.g., Suchman and Trabasso, 1966; Scandura and Voorhies, 1971). Indeed, in many situations, there are apt to be very good reasons for selecting such rules. Using one counting procedure might be satisfactory for adding small numbers, for example, but not for large ones.

In the next two sections, two basically different ways are considered for extending the partial theory of Chapter 7 to account for motivation. Specific attention is given to hypotheses which lend themselves to prediction as well as explanation. We also comment briefly on the problem of control, or how to manipulate motivation.

3. EXPERIENCE-FREE HYPOTHESES

In what we shall call experience-free hypotheses, no provision is made for individual differences. Such hypotheses are designed, not to deal with the behavior of individuals, but to provide an adequate account of (or to predict) statistics of group data.

This approach to theorizing is basically probabilistic in nature and does not lend itself to deterministic theorizing of the sort proposed in Chapter 7. Nonetheless, it must not be thought that experience-free hypotheses are somehow unusual. Indeed, it is precisely this sort of hypothesis which best characterizes contemporary theorizing in experimental psychology, theorizing which attains its clearest form in the stochastic theories of

---

6(cont'd.) (e.g., direct, constructive, indirect, etc.), or, more indirectly, as in estimating latencies. In the latter case, in contradistinction to what Suppes and Groen did, what might be done is to use, say, psycho-physical methods to obtain relatively stable estimates of the basic latencies (i.e., setting the counter and incrementing by one) under well-specified boundary conditions. (Suppes and Groen made no serious attempt to either specify or control miscellaneous factors which may have affected latency. As Groen pointed out in his dissertation, the subjects were not trying uniformly hard on all of the problems and tended to become more and more distractable as the experiment progressed.)

Then, after observing the subjects' performance on the test instances in question, the idea would be to simply identify which of the available procedures provides the best account of the data.
mathematical psychology. First, specific assumptions are made about how subjects learn, perceive, or remember. These assumptions, then, are typically converted into probability statements and predictions are derived which pertain to various group statistics.

This is a perfectly valid procedure as long as one recognizes that such assumptions are not really designed to reflect individual processes. It is instructive, therefore, to consider some specific experience-free hypotheses which might reasonably be proposed to account for rule selection.

Perhaps the first thought that comes to mind is the hypothesis that subjects tend to choose the path (rule) of "least resistance." This suggestion is not new and has been made in one form or another by any number of investigators (e.g., Bruner, Goodnow & Austin, 1956; Underwood & Schultz, 1960). The hypothesis, however, is inadequate as it stands. In order to make empirical predictions possible, paths of least resistance must be characterized in terms which can be determined independently of the selections subjects actually make. It would do no good to say that a particular rule is the path of least resistance simply because that is the rule the subject selects.

One possible hypothesis, for which some empirical support exists, is that where a number of rules of varying generality (cf. Scandura, Woodward & Lee, 1967; Chapter 2, Volume II) are available, subjects tend to select the one that is least general. For example, suppose that the subject "knows" the following summing formulas: sequential addition for summing arbitrary number series; \((A+L)/2\)N for arithmetic series; \(N^2\) for arithmetic series beginning with one and having a common difference of two; and the sum 2500 for the particular series 1+3+5+...+99. According to the above hypothesis, the subject would use: (1) the direct "association" (i.e., 2500) in responding to the series 1+3+5+...+99, (2) \(N^2\) in responding to, say, 1+3+5+...+67, and (3) \((A+L)/2\)N in responding to, say, 2+6+10+...+42.

Unfortunately, the literature often fails to make this fact explicit. One frequently finds, for example, statements which suggest that individual subjects perform according to assumption because theoretical predictions are supported by statistics of the group data. In actual fact, such hypotheses may generate predictions which are in close agreement with the obtained group statistics even where the hypotheses have little to do with how many of the subjects involved actually performed the task. A case in point is provided by the heated argument among many psychologists during the late 1950's and early 1960's over whether learning takes place in gradual increments or on an all-or-none basis. Both assumptions have been used to generate the traditional group learning curve. The Groen (1967) data, described in the previous section, further illustrates that even where the central concern is with individual behavior, the underlying theories are still based on averaging. In particular, the algorithm which provides the best overall account of a subject's data may have little or nothing to do with how he attacked any particular problem.
A variant of this hypothesis was proposed by Scandura (cf. 1966; Roughead & Scandura, 1968) to account for the results of a study on discovery learning. The recent results of Guthrie (1967), Worthen (1967), and Groen (1967) (see above) also provide indirect support for this contention. In the Roughead and Scandura study, for example, the subjects were given an opportunity to learn and use two procedures for generating formulas. One procedure was highly specific; the other was considerably more general. Those subjects who learned the specific procedures first did not bother to discover the more general procedure when given an opportunity to do so. On the other hand, those subjects who were given an opportunity to discover the more general procedure first did succeed in making the discovery. These studies probably involved more than rule selection, however, and thereby do not provide a direct test of the hypothesis. Indeed, the discovery subjects had to derive a new rule, rather than just select one. This was not true in the Groen (1967) study, however. In this case, assuming that specific rules are selected before general rules would account for the fact that many of Groen's subjects appeared to use more specific (rote) procedures in responding to certain familiar items, like 1+1 and 2+2. Most of the subjects were undoubtedly also able to apply algorithm (5) to these problems but declined to do so. Simply giving the learned response was easier.

It must be borne in mind that this is only one of any number of alternative hypotheses, which either separately or in conjunction might prove to be more useful. Among the more obvious possibilities is that complexity is a critical factor. Thus, for example, subjects might tend to select those available procedures (or paths) which contain the fewest atomic rules (in the sense of Chapter 7, Section 3).

Another possibility might be that subjects tend to select those rules which keep memory load (e.g., depth of postponement, Yngve, 1961) to a minimum. In doing mental arithmetic, for example, the usual arithmetical algorithms are not likely to be very helpful. The number of partial products, sums, and so on which need to be kept in short-term memory could easily overload most subjects' capacity for processing information (cf. Chapter 10).

4. EXPERIENCE-SENSITIVE HYPOTHESES

Experience-free hypotheses are based on the assumption that all sub-

---

8Gerald Satlow and Francine Endicott recently conducted an unpublished pilot experiment which generally favors this hypothesis.
jects select rules in the same way. Although characteristic of most behavioral theories, hypotheses of this sort are not compatible with the present approach to theory construction.

Instead of talking about how (all) subjects make selections, we allow for individual differences by introducing weaker assumptions to the effect that individual subjects make their selections in a systematic and predictable way. In view of the regularities frequently observed in the behavior of given individuals, this would appear to be a reasonable kind of assumption to make. The reaction of a scrooge to a poor waif at Christmas, for example, is likely to be quite different from that of a philanthropist, but the reaction of each is likely to be very much the same in similar kinds of situations.

Perhaps not so easily accepted is my belief that the bases for such selections are both determinable and subject to empirical manipulation. The main arguments proposed in its favor are of a largely theoretical nature, but some empirical support is reported in Section 4.2.

4.1 Mechanisms of Motivation

For some time, I was deeply puzzled about the best type of mechanism to introduce to account for motivation (rule selection). A number of alternative hypotheses and approaches were tried and a fair amount of pilot data were collected along the way with the help of others. One of my first thoughts was that given a class of rule-governed behaviors, different subjects may select different rules but that each subject selects the same rule consistently. As we shall see below, this Einstellung type assumption is similar to that made by Levine (1966). The pilot data reported in Scandura (1971b; Volume II, Chapter 7) indicated this to be a viable assumption but there were enough discrepancies to make me uneasy. In particular, some of our subjects tended simply to alternate in their selections on succeeding test instances.

This observation led me to suspect that rule selection itself might be a rule-governed activity—that is, that people use rules to select rules. In attempting to develop this idea, I at first thought that rule selections

---

Recall that the simple performance mechanism is based on the assumption that subjects select and test available rules one by one until one is found that satisfies the goal situation. This mechanism is perfectly adequate for predicting performance with respect to a given goal. It says nothing, however, about which rule will be used where there is more than one which applies in the goal situation. In turn, nothing can be said concerning any specific characteristics of responses that are attributable to specific rules.
might be made on the basis of specific characteristics of goal situations over and above those needed to determine whether or not a rule is applicable. In this case, the selection rules might be viewed as decision rules of the form "If (the stimulus has) property A (in addition to those required for rules $r_a$, $r_b$, $r_c$, etc. to be applicable), then select rule $r_a$; if not, and property B, then select $r_b$; if not, and property C, then select $r_c$; etc." In the case of alternation, for example, a distinction can be made between goal situations, depending on which rule was used on the previous instance. (The situation was thought to be like that described in Chapter 7, Section 5 on discovery learning where each goal situation includes previous S-R instances as well as the (nominal) stimulus itself. As we shall see below, the alternation rule requires memory of which rule was used on the preceding trial, but not of the S-R instance generated.)

In order to make predictions about selections, it was originally assumed that: Given a class of goal situations (involving the same goal), which can be satisfied by more than one rule available to a given subject, then the subject will use a single selection rule to make his selections in every goal situation in the class. This assumption has the effect of allowing different selections in a class of goal situations as long as the same rules are selected wherever the characteristics specified by a selection rule are identical. The subject is not allowed to select one rule in one goal situation and a different rule in another goal situation of the same type.\footnote{Distinguishing goal situations according to the rule used on the previous trial has the effect of imposing a partition on the Cartesian product set of the class (C) of rule-governed behaviors and the two-element set \{r was used on the preceding trial, r was not used on the preceding trial\}. (The Cartesian product $A \times B$ of sets A and B is the set of ordered pairs such that the first element is in A and the second in B (i.e., $A \times B = \{(a,b) \mid a \in A \text{ and } b \in B\}$).) This partition consists of two equivalence classes, one containing pairs consisting of S-R instances in C and "$r_1$ was used on the preceding trial" and the other containing pairs consisting of S-R instances and "$r_1$ was not used on the preceding trial." Notice that each instance in C is an element in two pairs.}

\footnote{Relaxing this requirement results in what have been called nondeterministic programs (cf. Chapter 2, Section 2.2). For a more rigorous treatment, see Scott (1967).}

\footnote{Notice that this hypothesis goes short of saying which rule the subject will pick in each equivalence class of goal situations. The experimenter may have particular selection rules in mind, which he uses to identify the goal selection criteria in the first place, but it is not essential to specify them in order to make predictions. For example, one such criterion might take into account which rule the subject used on the previous trial. However, although alternating behavior seems to be a particularly common phenomenon in two-choice situations, it is not essen-
All this was a step in the right direction but initially I was led astray by tying the problem of motivation in too closely with the practically useful idea of assessing behavior potential. Rather than keying on the mechanism itself, I was too concerned with how to explain the selections subjects make on particular rule-governed classes. This, it has turned out, was too restrictive a view.

I still feel that people use rules to select rules, but these selection rules are not necessarily tied to particular rule-governed classes (or, equivalently, particular classes of goal situations). They apply more generally in classes of classes of goal situations, in which each goal situation involves making an analogous choice among rules. Consider, for example, the general selection rule

(1) If \( r_1 \) was used on the preceding trial, then use rule \( r_2 \);
else use \( r_1 \) (where \( r_1 \) and \( r_2 \) are arbitrary rules).

Rule (1) leads to alternating behavior of the sort described above, and can be used to select rules in a broad class of classes of goal situations. In particular, it may be involved in any goal situation which can be satisfied by either of two rules.

This generalization not only makes it easier to explain the process of rule selection but overall parsimony of the theory increases as well. In effect, one general mechanism seems to suffice for rule selection as well as for learning (problem solving) and simple performance. Given a goal situation, control automatically shifts to the higher order goal of generating (i.e., selecting or deriving) a rule which may satisfy the goal. Conversely, if the output rule generated by a higher order rule satisfies a higher order goal, then control reverts to the (next) lower level (if any) and the output rule is applied. One final assumption is necessary in order to keep control from shifting to higher order goals ad infinitum:

\[ \text{(cont'd.)} \]

The internal consistency of a theory has in the history of science often proved to be a positive sign, especially where the theory deals with a broad range of phenomena and correspondingly makes contact with observables in a number of distinct ways. As often as not, empirical findings have played a distinctly secondary role in the formulation of some of our more useful and long-lived theories. An often-referred-to example of this was in relativity theory, where Einstein was motivated as much (or more) by considerations of internal consistency as by relevant empirical results such as those of Michelson and Morley on the constancy of the speed of light. At any given point in time there is always more potentially relevant data available than can be mastered by any one individual. The crucial problem is selection.
If exactly one rule satisfies a given higher order goal, then control shifts down and the rule is applied. Without such an assumption, the subject could get caught in an infinite regress trying to find a level at which he might apply a rule and would therefore be unable to act. (Perhaps this is what Miller, Galanter, and Pribram (1960, 8-10) had in mind when they criticized Tolman's rats for being so engrossed in thought that they could not perform.)

Given any goal situation, then, according to this mechanism control initially reverts to the higher order goal of generating a rule which may satisfy the original goal. Control continues to shift upward in this way until exactly one rule may work. At this point, control reverts to the next lower level, and the rule is applied. If the new rule generated satisfies this lower level goal, control moves down another level and the generated rule is applied. This continues until the original goal is satisfied. Notice that selection rules may be involved wherever more than one rule satisfies some goal in the hierarchy.

Assume, for example, that the subject has available the alternating selection rule (1) above and the two rules, \( r_1 = 1/2 \text{ bc } \sin A \) and \( r_2 = \sqrt{s(a-b)(s-c)} \), for finding areas of triangles where \( a, b, \) and \( c \) are sides, \( s = 1/2 (a+b+c) \), and \( A \) is the angle opposite side \( a \). Also suppose that the subject has been presented with a sequence of goal situations involving finding areas of triangles given the length of their sides \( a, b, \) and \( c \) and angles \( A \) opposite sides \( a \). Finally, assume that the rule last used was \( r_1 \). Then, given a new goal situation of this type, control automatically moves to the higher order goal of finding a rule which satisfies the initial goal situation. Since \( r_1 \) and \( r_2 \) both satisfy this higher order goal, control moves to the still higher order goal of finding a rule for making a selection. Here, only rule (1) satisfies, so control reverts to the next lower level and it is applied yielding rule \( r_2 \) as output. Rule \( r_2 \) satisfies the higher order goal so control reverts to the initial goal and \( r_2 \) is applied and the problem solved (i.e., the output of \( r_2 \) satisfies the initial goal).

This mechanism also suffices for learning (problem solving) and simple performance. Learning follows in the same general way as in rule selection. Simple performance is just a variant of rule selection in which we never have to go beyond the (first) higher order goal because this goal is satisfied by exactly one rule. The original performance and learning mechanisms of Chapter 7 work perfectly well, of course, where we do not care which of the available rules that satisfy the (first) higher
Although appealingly parsimonious, this mechanism unfortunately does not take all possibilities into account. In particular, it does not allow for "false starts." Any complete account of behavior must allow for the possibility of generating rules which satisfy a particular higher order goal but which when applied generate outputs which do not satisfy the next lower order goal. The above mechanism does not provide for this case and implicitly assumes a direct connection among the lower and higher order rules.

Consideration of false starts here would be unwieldy, and discussion is postponed until Chapter 9 where the theory is more rigorously formulated. For the present, it is sufficient to note that whenever a false start occurs, control is assumed to again shift upward and another available alternative selected. In the absence of relevant data, it is unclear as to the exact mechanism for selecting alternatives, but one possibility would be for control to shift upward until the first previously used selection rule (if any) is met. Control would then revert to the next lower goal and this selection rule would be applied once more; only this time, the rule selected would be the first alternative to that selected on the previous go-around. This new output rule would also satisfy the (next lower) goal and, in turn, would be applied.

Before moving to a direct test of the motivation mechanism, there is one additional problem to be disposed of: How are we to determine which of the alternative rules the subject actually uses? The initial goal criterion presented to the subject will not suffice since this criterion is satisfied by each of the available rules. We could ask the subject, of course, but this makes the observer dependent on the subject

---

It should be cautioned that the proposed hypotheses, like those described in Chapter 7, are applicable only under very special circumstances. In particular, they apply only where the situation is highly structured—that is, where it is clear what the goal and available rules are and where the subject is unencumbered by memory. In experimental testing, extreme care must be taken to meet these boundary conditions. For example, one would not expect these hypotheses to apply directly to relatively unstructured classroom situations—say, where the subject is presented with a choice of different activities. The major source of difficulty in such situations would be that of identifying a suitable competence (knowledge) model. Expecting precise predictions under these conditions would be much like requiring that a physicist be able to predict where a given leaf will fall on a windy day on the basis of his knowledge of aerodynamics.

It is for exactly such reasons that it is not possible in principle to falsify any scientific hypothesis (Ossorio, 1968). The ultimate value of the proposed hypotheses will depend on their range of applicability and, particularly, on the importance of the implications which can be derived from them, and possibly other hypotheses in the theory.
for confirmation of his predictions.

Our preferred alternative is to introduce response criteria which adequately distinguish among the various rule alternatives. Clearly, this will be easier to do in some cases than in others. According to the discussion in Section 2, one might suspect that distinguishing among rules for adding numbers would require more exacting (additional) criteria than distinguishing among rules for proving theorems, or building houses. The former, for example, might require accurate latency measures whereas the latter might just require determining the type of proof (e.g., direct, indirect), or type of house (e.g., brick, shingle, etc.).

4.2 Selection Rule Study

The following experiment was conducted jointly with Francine Endicott and provides a direct test of the selection mechanism.

Method

Tasks and Materials. The experimental tasks were essentially those used in the Chapter 7 experiment involving the trading games. In each task, \( n \) input objects (e.g., pennies) were traded for \( n + m \) output objects (e.g., candles), where \( n + m \leq 12 \) with \( n \leq 7 \) and \( 2 \leq m \leq 5 \). These restrictions helped to insure both ease of computation and sufficient variety.

Two kinds of rule were used in the study; both were variants of the simple rules used earlier. One kind was called "Match first, then add" (MA). MA rules involve: (1) taking output objects (from a supply bowl) and forming a one-to-one correspondence with the \( n \) input objects and (2) adding \( m \) more output objects. In the other kind of rule, "Add first, then Match" (AM), these steps were reversed. That is, AM rules involve: (1) adding \( m \) more input objects to the \( n \) input objects and (2) forming a one-to-one correspondence between the output objects and the \( n + m \) input objects. The two kinds of rule are illustrated below for a trading game task, with \( n = 3 \), \( m = 2 \). The \( \bigotimes \) refer to given inputs, \( \bigcirc \) to added inputs, and \(-\) to outputs.
In addition to these two kinds of rule, four selection rules were taught. (Selection rules, recall, are analogous to the composition and generalization rules used in the problem solving studies.) The four selection rules were represented on cards as follows:

**Alt-MA** Alternate beginning with MA.

<table>
<thead>
<tr>
<th>First use</th>
<th>Match</th>
<th>Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>then use</td>
<td>Add</td>
<td>Match</td>
</tr>
<tr>
<td>next use</td>
<td>Match</td>
<td>Add</td>
</tr>
<tr>
<td>next use</td>
<td>Add</td>
<td>Match</td>
</tr>
<tr>
<td>and so on.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Alt-AM** Alternate beginning with AM.

<table>
<thead>
<tr>
<th>First use</th>
<th>Add</th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>then use</td>
<td>Match</td>
<td>Add</td>
</tr>
<tr>
<td>next use</td>
<td>Add</td>
<td>Match</td>
</tr>
<tr>
<td>next use</td>
<td>Match</td>
<td>Add</td>
</tr>
<tr>
<td>and so on.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**C-MA** If the input is candy, use MA. Otherwise, use AM.

<table>
<thead>
<tr>
<th>If candy, use</th>
<th>Match</th>
<th>Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>If not candy, use</td>
<td>Add</td>
<td>Match</td>
</tr>
</tbody>
</table>

**C-AM** If the input is candy, use AM. Otherwise, use MA.

<table>
<thead>
<tr>
<th>If candy, use</th>
<th>Add</th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>If not candy, use</td>
<td>Match</td>
<td>Add</td>
</tr>
</tbody>
</table>
Subjects, Design and Procedure. The subjects were 12 second and third grade pupils in one private and one public school in West Philadelphia. They were tested individually and were given a piece of candy at the end of each experimental session.

At the beginning of the experiment, the experimenter said, "We're going to play some trading games. I'll teach you two ways to play the trading games. Here's a trading game problem (task). I put out \( n \) of these inputs. (In the actual instructions, the italicized terms were replaced with specific names—e.g., dolls.) You have to give me the right number of these outputs." Then the experimenter taught the subject how to solve the tasks, first using one rule (AM or MA) and then the other. The order of presentation of the two kinds of rule was randomized over subjects. A sign was placed just behind the input and output objects to indicate which rule was to be used on a given trial.

Each subject worked several problems using both kinds of rule. The experimenter helped the subject to see that he got the same number of output objects, irrespective of whether he used rule AM or MA. The experimenter did this by asking such questions as "How many outputs did you give me?" each time the subject solved a problem, and by observing that the number of outputs was the same no matter which kind of rule was used.

This procedure was continued until the experimenter was satisfied that the subject could correctly use the rules indicated by the signs and that the subject was aware that both kinds of rule yielded the same number of output objects. At this point, additional practice was provided on several new trading game problems. On each practice problem the rule to be used was stipulated (as indicated above).

A pretest was given next. It consisted of three sets of new trading game problems of four, three, and two problems respectively. The subjects were instructed

"Now I will give you three sets of trading game problems. Each of the two kinds of rules you have just learned will work on these problems. I will put out a problem. And I will put out two rule signs." The experimenter added, "Before you work the problem, pick up and place here the sign of the rule you will use to solve the problem. There is a way to pick which kind of rule to use on each problem. Let's see if you can guess which kind of rule I want you to use."\(^{15}\)

\(^{15}\) In retrospect it is questionable whether we should have told the subject to actually pick up and place the rule signs. This made it easier to identify which rule the subject was using but had the effect of stating the higher order goal directly (although not how to achieve it).
The subjects were randomly assigned to one of four selection rule treatment groups. The experimenter presented the card describing the indicated selection rule and told the subject: "On this card are the directions which tell you how to choose between the two kinds of rule. From now on follow these directions." Then the experimenter explained how to use the selection rule. To confirm that the subject understood, the experimenter presented three sets of problems consisting of four, three, and two problems respectively. The subject was told to simply indicate which kind of operation rule should be used, but not to work the problem. Where necessary, the experimenter corrected the subject and re-explained the selection rule.

The posttest consisted of three new sets of problems of four, three, and two problems, respectively. The instructions were "Now I will give you three sets of trading game problems. Each of the two kinds of rules you have learned will work on these problems. I will put out a problem. And I will put out two rule signs. Before you work the problem pick up and place here the sign of the rule you will use to solve the problem. Choose which rule you use on each problem according to the directions on this card."*

The experimental procedure was conducted over two experimental sessions. The first session included the procedure up to the pretest. The second session began with a practice-review of the AM and MA rules.

Results

The results for all 12 subjects are summarized in the Table below—including any selection rules that appeared to be operative during the pretest, the selection rules taught (treatment), and the selection rules that appeared to be operative during the posttest.

Generally speaking, the results were consistent with the hypothesis in all 12 cases. It is instructive, nonetheless, to consider the deviations. During the selection rule practice and on the posttest, Subject #4 seemed to have difficulty remembering what problem she was on, and because she received the Alt-MA treatment, had some trouble determining which operation rule to use. On the fourth problem of the first set of posttest problems, it appeared that she had lost track of where she was, and she picked AM instead of MA. Subject #3, on the other hand, had no such difficulty; he seemed to simply note which rule sign was up from the last problem and chose the other sign. After Subject #4 had been run,

*The posttest instructions were somewhat more directive than in the comparable studies reported in Chapter 7. This was necessary to insure that the subjects did use the selection rules taught rather than the more idiosyncratic selection rules they were accustomed to using in (cont'd.)
the procedure was changed slightly to emphasize this method with the remaining subjects who were taught an "alternate" selection rule.

Subject #7, on the first problem of Set 3 on the posttest, chose MA instead of AM as he was expected to. The problem involved erasers as input and flags mounted on toothpicks as output. This was the second time in the posttest that the erasers–flags situation had appeared. At the end of the first appearance of the erasers and flags he discovered that he could stick the flags into the erasers. On problem 1 of Set 3, then, he seemed distracted by the prospect of sticking the flags into the erasers, and grabbed the wrong rule sign.

Over all, there were 105 or 97% correct predictions out of 108. Based on the assumption that correct and incorrect predictions are binomially distributed, the 87% confidence interval for correct predictions was between 94% and 100%.

Subject Pretest Treatment Posttest
1 ? C-MA C-MA
2 AM C-AM C-AM
3 ? Alt-MA Alt-MA
4 ? Alt-AM Alt-AM 8/9
5 ? C-AM C-AM
6 Alt Alt-MA Alt-MA
7 ? Alt-AM Alt-AM Sets 1 & 2
     Alt-MA Set 3
8 ? Alt-MA Alt-MA
9 ? C-MA C-MA
10 MA Alt-AM Alt-AM
11 Alt-MA Set 1
   AM Set 2
   Alt-AM Set 3 C-AM C-AM
12 Alt-MA Set 1
   ? Sets 2 & 3 C-MA C-MA

4.3 Further Analyses and Implications

If both the relevant lower order rules and the selection rules available to a subject are known, the motivation mechanism makes it possible in principle to predict his behavior perfectly. Unless these rules are built directly into the subject, however, we can never know for sure the full extent of his capabilities; and our predictions, accordingly, become *(cont'd.) such situations. In Chapter 7, recall, the subjects did not know any adequate higher order rule prior to training. In more structured situations, the selection rules used are considerably fewer in number and easier to identify (cf. Section 4.3).*
less certain. This circumstance is not as restrictive in practice as one might suspect—for many of the same reasons that it is not necessary in assessing behavior potential to know precisely which rules a given subject has available.

Selection Rule Unknown. In practice, there are several kinds of partial knowledge that an observer might have. Given a particular goal, he might, for example, know which relevant lower order rules are available to a subject, but not know the available and relevant selection rules. In order to predict a subject's selections, the observer must take into account as many different selection rules as the subject might possibly use. The basic problem is to determine which one or what combination of these selection rules controls the subject's selections.

To see how this might be done, first note that each selection rule partitions the class of to-be-selected alternative rules according to certain criteria. Consider, for example, the alternation rule: "If \( r_1 \) was used on the preceding trial, then select \( r_2 \); otherwise, select \( r_1 \)." (This selection rule applies to a broad class of rule pairs, denoted \( \{ (r_1, r_2) \mid r_1 \text{ and } r_2 \text{ satisfy the same class of goal situations} \} \).)

In principle there are any number of selection rules that a given subject might use. But, in practice, the number of viable ones with structured tasks will generally be fairly small.

The criteria upon which the various selection rules are based generally will be independent of one another in the sense that, given a particular goal situation in which two or more selection rules may be involved, the outputs of these selection rules may or may not be identical. In one case, for example, suppose that the selections are based on the relative number of components (in the alternative rules), while in another, the selections are based on which rule was used on the preceding trial. Clearly, the number of components in the two alternative rules is independent of which rule was used on the preceding trial. In effect, each of the alternative selection rules may impose a different partition on the class of (sets of) alternative rules.

Nonetheless, within each element of the intersection\(^{16}\) of such partitions, selections are necessarily identical no matter which selection rule is used. Assuming that the selection rules are exhaustive and sufficiently fine grained, the choices a subject makes within the equivalence classes will be consistent, according to our theory. Although the actual selection

---

\(^{16}\)The term "intersection" is interpreted here as in Chapter 7, Section 3, and Chapter 9 rather than as a simple operation on sets.
rules being used by subjects may be unknown to the observer, it should be noted that it is always possible to construct single selection rules (there may be more than one) which correspond directly to the intersection partition, just as in assessing behavior potential. Such selection rules provide a basis for identifying selection rules which are behaviorally equivalent to the rules actually governing each subject's selections.

Once having determined the kinds of selections a subject is apt to make (or, equivalently, a selection rule which accounts for them), it is possible to make predictions concerning the subject's performance in associated goal situations where any one of two or more available and alternative rules might be used. Our major concern in this case is in predicting which of the refined goal criteria will be satisfied by the subject's responses. The motivation mechanism makes this possible by specifying which of the alternative rules is selected. Thus, for example, if the subject selects rule A, rather than, say, rule B or rule C, then according to the theory the response would satisfy the refined goal criteria associated with rule A. (In general, the same selection rule may apply to any number of different sets of alternative rules, but concern may be restricted to just one set of alternative rules—for instance, where the choice is between two rules for computing areas of triangles.)

Although the amount is not large, there is some relevant research in the literature which bears on this analysis (e.g., Hunt & Hovland, 1960; Levine, 1966, 1969) in the case where the selections are restricted to a single rule-governed class. The Levine (e.g., 1966, 1969) studies, in particular, provide a fairly direct test of a simplified version of these ideas in the restricted context of discrimination learning. Recall from Chapter 7 that Levine and his co-workers were concerned with the laws governing the choices subjects make on the various trials of a discrimination learning task. The basic approach they followed was to identify a number of feasible hypotheses (i.e., rules) which subjects might use to generate their responses on the various trials. At the start, all of these rules are consistent with the effective goal. Then, as a result of response feedback on the "learning" trials, the effective goal changes so that certain of the rules may no longer be applicable. To determine the subject's choice of rule on each learning trial (where reinforcement was given), the subject was presented with a sufficient number of nonreinforced (blank) trials to identify uniquely which of the identified and available rules was selected. Four were required with the task Levine (1966) used.

Based on this assessment, Levine was able to predict his subjects'
responses with 97.5% accuracy. His "learning" data are also generally compatible with the proposed mechanism (See Chapter 7, Section 5), but they are not as clear cut. The experimental conditions Levine has used in his research are not entirely memory-free and, hence, some deviation is to be expected. The more complex the task, the greater the expected deviation. In his more recent work, Levine (1969) has attempted to deal with the problem of memory by assuming that the subject can consider at most a small finite number of hypotheses at any one time.

Although Levine's theory, at least the memory-free parts of it, is compatible with that proposed here, it is important to note that certain terms are used differently in each. In addition to calling rules "hypotheses," which is not likely to cause any confusion, what Levine calls "learning" is not learning in the sense described in Chapter 7. No new rules are acquired. The subject merely determines, through feedback, which of the available rules the experimenter wants him to use. In short, it is a question of rule selection.17

The major limitation of the Levine research is that the underlying theory is framed in too restrictive terms, particularly insofar as the basis for selection is concerned. Levine effectively assumes that the subject will make the same choice of rule, at each stage of "learning," in all goal situations. It is, in effect, an Einstellung type assumption. Our analysis allows for any finite number of different choices.18

Selection and Lower Order Rules Unknown. A second kind of situation in which an observer might find himself is that of knowing neither the selection rule nor the specific lower order rules which the subject has available for dealing with a given goal situation. In this case, the observer has to make intelligent guesses not only as to the kind of selections the subject is apt to make, but also the kind of rules he is apt to have available for achieving the goal. He must also be in a position to judge which components of these rules may appropriately be thought of as atomic.

The major task here is to determine through a testing procedure of some sort (1) which paths of the hypothesized lower order rules given subjects have available and (2) where two or more paths lead to the same

---

17It is also worth noting that Levine's (1969) recent attempts to bridge the memory gap correspond roughly to the type of enriched structure introduced in Chapter 10.

18Do not confuse the alternating pattern of responses often obtained in Levine's studies (e.g., 1966) with the alternating pattern of rule selection referred to above. Levine's alternating response patterns are generated by single rules.
goal, which selections the subjects make. Suppose, for example, that a third-grade pupil is presented with pairs of numbers and is asked simply to compute. Under these circumstances, the child might add, subtract, multiply or divide and still satisfy the goal criterion. With certain pairs of numbers, of course, he might not be able to perform any of these computations. Here, the experimenter-observer's job is to determine, first, which pairs of numbers the child is able to compute with and, second, where he can compute, whether he will give the sum, difference, product, or quotient. 19

In this case, it is necessary to determine which paths of the various computational rules the subject has available and, where more than one path applies (in a given goal situation), which one he will select. In dealing with situations of this sort it is sometimes convenient, even if not necessary, to restrict selection rules to the rule-governed class under consideration.

The first step is to form the intersection of the partitions associated with the various lower order rules introduced as possible candidates for generating the behavior. The procedure follows directly from that described in Chapter 7, Section 3. Call this the rule intersection partition. A simple rule intersection partition is represented schematically in the figure below by the horizontal and vertical solid lines. (Ignore the other lines for the moment.) To make things definite, each

---

19 Although this task may appear to be somewhat contrived, it was picked solely for convenience. The same basic questions would apply to any situation where the responses generated by two or more rules can be readily distinguished.
$S_i P_j (P_j S_i)$ where $i, j = 1, 2$ may be thought of as the intersection of equivalence classes corresponding to path $S_i$ of a summing rule and path $P_j$ of a multiplication rule. Stated differently, the responses corresponding to the intersection equivalence class $S_i P_j$ can be generated by either path $S_i$ or path $P_j$. Both paths generate responses which satisfy the initial goal conditions.

Second, the observer must identify as many selection rules as he feels are necessary to (essentially) exhaust those that the subject in question is likely to use. He then forms the corresponding selection intersection partition on the rule-governed class in question. (This process is illustrated in footnote 10 on page 267 for the alternating selection rule restricted to a given rule-governed class.)

Without loss of generality, the (inner) circle in the figure can be thought of as the boundary of the partition resulting from the alternating selection rule. In particular, we assume that the responses associated with the interior of the circle are generated in goal situations where an addition has been performed on the preceding trial, and the exterior, to goal situations where a multiplication has been performed. This has the effect of distinguishing S-R instances according to the rule used on the preceding trial. Hence, strictly speaking, the above figure is a Cartesian product—cf. footnote 10 on page 267. (As an exercise, consider the partitions imposed by the other selection rules used in the experiment reported in Section 4.2. Notice the simple form the partitions take.)

The solid lines in the figure, then, represent the intersection of the rule intersection partition and the intersection partition imposed by the selection rules. According to our theory, subjects would be expected to select the same path, if any are available, in each goal situation corresponding to any given equivalence class of the partition. This follows because the selection criteria are identical within each such equivalence class. (Although the intersection partition is defined on a given class of rule-governed behaviors, it can naturally be thought of as a partition on the class of corresponding goal situations where each instance corresponds to a unique goal situation and vice versa.)

Which paths a subject selects, if any, are determined by testing him in one goal situation in each equivalence class of the (final intersection) partition. Rather than observing which paths are selected directly, of course, predictions will in most cases be tested against refined response criteria imposed by the observer to distinguish among the various possible paths. In the computation example, the experimenter must find
criteria which distinguish sums and products. The partition corresponding to the refined criteria is illustrated in the above figure by diagonal broken lines.\textsuperscript{20} The shaded areas indicate one possible set of outcomes (i.e., refined criteria which are satisfied as a result of testing).

Future predictions, then, are made in accordance with the refined goal criteria the particular responses satisfy. According to assumption, subjects will select the same path in any goal situation corresponding to a given equivalence class.

Notice that this approach is essentially a generalization of that proposed in Chapter 7, Section 3 for assessing behavior potential, and makes it possible in principle to predict for each stimulus which of any finite number of responses a subject will give.\textsuperscript{21} In many situations, however, determining what might be called multiple behavior potential could require a relatively large number of test instances and become unwieldy. Fortunately, the number of such instances can be reduced in certain cases. The more information the experimenter has about what rules a subject knows, for example, the better position he will be in to eliminate unnecessary equivalence classes in the rule intersection partition. Information about selection rules may similarly reduce the number of equivalence classes and, hence, the total number of test instances required. Furthermore, in practice, where unknown selection or lower order rules are likely to be involved, the analyst may choose to concentrate only on those alternatives of special concern, and to introduce a complementary subclass of responses to allow for "errors."

Of somewhat more theoretical interest is the possibility that inherent relationships among the various equivalence classes may provide an adequate basis for various, relatively efficient, forms of conditional testing. Since the paths associated with the rule intersection partition can be

\textsuperscript{20}It is important to emphasize that the refined criteria must be compatible with the paths the observer wants to distinguish. Suppose, for example, that, instead of distinguishing between sums and products directly, the experimenter had arbitrarily imposed a latency criterion, say, of three seconds. On the average, subjects might give sums faster than products but this would hardly be true in every case. To this extent, the latency criterion would be inadequate insofar as distinguishing between sums and products is concerned.

The difference between this simple latency criterion and that used by Suppes and Groen (1967) is that in the latter case the latency criteria used in each goal situation were based directly on the hypothesized rules (paths).

\textsuperscript{21}In the approach described earlier remember it is only possible to say which items a subject can respond to correctly and which he cannot.
partially ordered (cf. Chapter 7, Section 3), for example, one can start by testing as before on items at the highest levels of the lattice. Success on any higher order path effectively implies that the equivalence classes associated with the relatively lower level paths can be combined to form one large equivalence class. In short, testing can stop at the higher level, thus reducing the number of test instances needed. Carrying out this type of testing in practice might best be done with the aid of a computer, or in the absence of that, at least a well-prepared and nimble-minded experimenter.

**Probabilistic Selection.** Useful predictions may also be obtained in relatively unstructured situations by modifying the motivation mechanism to allow for probabilistic, rather than deterministic, selections. Consider, for example, the ordinary classroom where we may want to make predictions concerning the various choices among activities that individual students continually have to make—choices, for example, between baseball and bull sessions during free play or painting and sculpture during art class. In situations of this sort it would be foolhardy to attempt completely definitive accounts. The goal situations may vary ever so slightly and in unknown ways from choice to choice so that a deterministic approach might be totally impractical.

Nonetheless, the probabilities involved in this type of situation may be sufficiently stable as to make useful predictions possible. Anyone who has worked with children, for example, knows that they have very definite preferences for the kinds of activities in which they participate. Furthermore, there are some situations, at least, where the kinds of choices made by particular individuals tend to fall into a regular and predictable pattern. The existence of such patterns of behavior is the source of such common remarks as "George always takes the hard way" or "Vivian is always so cooperative." In the school setting such tendencies frequently manifest themselves in the more or less ready acceptance of challenging, as opposed to routine, intellectual activities.

Assuming that such preferences tend to remain stable, at least over short periods of time, it may prove possible to predict (or at least explain) the kinds of choices students make in one type of choice situation on the basis of the kinds of choices made in analogous situations.

There is a large body of related literature on utility theory (e.g., Luce and Suppes, 1963) but to my knowledge there are no data which bear directly on this issue. There are a few studies, however, which deal in various ways with selections among two or more available rules. Wittrock and Hill (1968), for example, have shown that knowing
The preferences of young children can frequently be used to advantage in predicting success on transfer tasks. There are also some data (Suchman & Trabasso, 1966) which suggest that the selection rules used by young children (i.e., their preferences) may change over time in predictable ways. Specifically, Suchman and Trabasso found that young children prefer color while older ones prefer form.²²

²²This suggests that preferences (i.e., selection rules) might be manipulated in the school setting, presumably by various combinations of reward, punishment, and/or nonreinforcement. Indeed, this is undoubtedly being done now. A greater awareness of the principles involved, however, not only might lead to improved practice, but also might provide a more explicit basis for more sophisticated educational technologies which take student motivation directly into account.
In the theory of knowledge $\mathcal{K}$ the theorist's task is to devise a finite rule set $K$ which accounts for a given class $A: B$. In order to have behavioral relevance, each rule known to a given subject(s) and relevant to the given class $L: B$ must be embedded in some rule in $K$ in a sense made explicit below. Put differently, $K$ must include all of the rules the subject or subjects are likely to know.

The idealized (memory-free) theory of structural learning $\mathcal{L}$, then, begins where $\mathcal{K}$ leaves off. There are two major additional problems with which the behavior theorist must deal. First, he must have some way of identifying the knowledge had by the subject in terms of that identified in his role as observer (qua competence theorist). Second, the theorist must have some way of explaining and predicting the subject's learning and performance.

The basic approach rests on the following assumptions:

(1) The behavior theorist either knows or can manipulate the subject's goal (i.e., what he is trying to do) with some known degree of specificity; where this is not the case the theory can only be used for explanatory purposes, not prediction.\(^1\)

\(^1\)Because this theory applies only where all of the relevant rules are immediately available to the subject, several people have suggested the label "Memory-Available."

Such an approach has been criticized (e.g., Witz, Chapter 4, Volume II; Knifong, 1971) as not providing a "dynamic" model which directly reflects observed behavior. In reaction, I would simply point out that the present approach provides a basis not only for explanation (cf. Piaget in Flavell, 1963; Furth, 1969; Witz, Chapter 4, Volume II) and description (e.g., Knifong, 1971), but also for prediction, something which the so-called dynamic models have been unable to provide. Indeed, although I shall not attempt to justify my belief, I suspect that the (continued)
286 Structural Learning

(2) Performance is goal directed. This assumption provides a basis for determining which (parts) of the rules in K individual subjects know.

(3) Learning and motivation depend on shifting control between given and higher order goals in a specified manner.

The formalism which follows involves these assumptions. For convenience, we limit our examples to \( \mathcal{L} : \mathcal{R} \).

1. GOAL SITUATIONS

Goals provide the theorist-observer with criteria for judging behavior. They are also assumed to control the subject's behavior. Definitions 1 - 4 make these notions precise.

Let \( \text{Dom} \) be a class of inputs (I) or stimulus situations (S) and \( \text{Ran} \) be a class of outputs (O) or behaviors (B). \( P \) is a binary predicate which defines a nonempty binary relation \( p \) in \( \text{Dom} \times \text{Ran} \).

**Definition 1:** A goal \( G \) is a class \{O | (I,0) \in p \text{ for some } I \in \text{Dom} \}

All goals can be described by an imperative statement of the form: "Find a behavior \( B \) such that \( B \) satisfies goal criterion (predicate) \( P(S,B) \) for some \( S \in \text{Dom} \)." Equivalently, "\! B, P(S,B)."

Notice that, whereas goals may be defined in terms of nonobservables (e.g., \( I, O \)), goal descriptions are strictly observable. Interpreting goal descriptions yields goals. For example, the goal \{O | O = a+b where \( a, b \) are natural numbers\} may be described "Find a response \( R \), such that \( R = a+b \) for the pair of natural numerals \( a, b \)."

By themselves, goals are insufficient for determining performance. They tell where one is going but not where one is. Stimulus situations or, more generally, inputs, provide the occasion for responding and together with goals give

**Definition 2:** A goal situation \((I_o, G)\) is a class \{O | (I_o,0) \in p \text{ where } I_o \in \text{Dom} \}

Notice that goal situations can be characterized as pairs consisting of an input \( I_o \) and a goal \( G \). All goal situations can be described by a statement of the form: "Find a behavior \( B \) such that \( B \) satisfies \( P(S_o,B) \) where \( S_o \in \text{Dom} \)." In practice, of course, goal situations may be described in a variety of equivalent ways. The statement "Find 3 + 4," for example,

\( \text{(cont'd.)} \) dynamic models described by Witz, for example, will turn out to correspond to situations where the observer knows relatively little about what the subject is trying to do ahead of time, and can only determine this after the fact.
is usually interpreted to mean "Find n such that n = 3 + 4."

In addition to the stimulus $S_0$, we may on occasion also want to refer to other stimulation in the environment, possibly including programs for rules. In this case we use the term environmental complex $E$.

**Definition 3:** A goal environment is a pair consisting of a goal situation $(I_0, G)$ and an environmental complex $E$.

Clearly, there is a close relationship between $\mathcal{dm}$'s and goals. The former are partitions whose elements are categories (sets) of outputs. The latter are (single) categories containing just those outputs which "satisfy" a goal. More precisely, the relationship is specified by a map $t$ in

**Definition 4:** If $\mathcal{dm}$ is a decision making capability, then $t_n(\mathcal{dm})$ is that goal given by the $n$th category in $\mathcal{dm}$.

For example, $t_1(\{R | R = 5 + 4\}, \{R | R \neq 5 + 4\}) = \{R | R = 5 + 4\}$ and $t_2(\{A = \{x | (5,4) \in \text{Dom } r, \text{ and for all } R \in \text{Ran } r, R = a + b \text{ for some } (a,b)\}, \bar{A} \text{ (i.e., complement of } A)\}) = \bar{A}$.

There are two senses in which a response may satisfy a given goal situation. The goal itself requires only that $B$ be a member of a particular class (e.g., the class of sums $a + b$). We call this the a priori criterion of the goal situation. This criterion defines only the range (type) of behaviors allowed and is independent of the particular stimulus. The stronger sense of satisfying a goal situation depends also on the stimulus (e.g., $B = 3 + 4$). In this case, the criterion involved is referred to as stimulus dependent. This distinction between a priori and stimulus dependent criteria plays an important role in the learning and performance mechanism described below.

2. NATURE OF THE IDEALIZED (MEMORY-FREE) THEORY

A memory-free theory of learning $\mathcal{L}$ may be defined as an $n + m + 3$ tuple $\mathcal{L} = \langle \mathcal{A}, \mathcal{B}, K, r_1, \ldots, r_n, S, \langle S_1, G_1 \rangle, \ldots, \langle S_m, G_m \rangle \rangle$ where $\mathcal{A}, \mathcal{B}, K, r_1, \ldots, r_n$ are as in $\mathcal{K}$, $S$ is a behaving subject, and $\langle S_1, G_1 \rangle, \ldots, \langle S_m, G_m \rangle$ are goal situations.

If the theory is to have predictive value, then the goal situations must be capable of manipulation and equally meaningful to both $S$ and the observer. Our main task in this section is to show how $\mathcal{L}$ provides a

---

3Obviously, if $B$ satisfies a stimulus dependent criterion, then it necessarily satisfies the corresponding a priori criterion.
basis for determining what \( S \) knows and how \( S \) performs and learns.

First, we introduce the notion of path of a rule and show how the paths of a rule impose a partition on the rule's extension.

**Definition 5:** A (completed) *computation* of stimulus \( S \) by rule \( r \) is the finite sequence \( S = S_1, o_{r_1}, 0_1, o_{r_2}, 0_2, o_{r_3}, 0_3, \ldots, o_{r_n}, 0_n = B \) (where \( S \) is in the domain of \( r \) and \( 0_i \) is the output generated by the preceding atomic operating rule \( o_{r_i} \)) obtained by applying in turn the atomic operating rules of \( r \) to \( S \). Notice that \( dmc's \)'s are involved in determining which \( o_r \) is applied next but do not appear in the computation itself.

**Definition 6:** A *simple path* of a rule \( r \) on stimulus \( S \) is the finite sequence of atomic operating rules \( o_{r_1}, o_{r_2}, \ldots, o_{r_n} \) of \( r \) obtained by deleting the stimulus and outputs from the computation of \( S \) by \( r \).

Intuitively, the simple path associated with any given stimulus \( S \) and rule \( r \) consists precisely of those atomic operating rules of \( r \) which are applied to produce the corresponding output, and in the order in which they are applied (including repeats).

**Definition 7:** The set of \( S-B \) pairs associated with a simple path of rule \( r \) is the class of \( S-B \) pairs that the simple path computes.

**Definition 8:** A *path* (form) is determined from a simple path by eliminating consecutive repetitions of subsequences. (Each repetition may be replaced by \( * \).)

For example, the simple path \( o_{r_1}, o_{r_2}, o_{r_2}, o_{r_3}, o_{r_4}, o_{r_3}, o_{r_4}, o_{r_2}, \ldots, o_{r_4}, o_{r_2}, \ldots, o_{r_4}, o_{r_5}, o_{r_6} \) is replaced with the path \( o_{r_1}, (o_{r_2}^*, (o_{r_3}^*, o_{r_4}^*)^*), o_{r_5}, o_{r_6} \).

**Definition 9:** Two simple paths are *equivalent* if they generate the same path form (up to \( * \)).

**Theorem A:** "Is the same form as" is an equivalence relation on the set of simple paths associated with a rule. Further, this equivalence relation *partitions* the extension of the rule.

**Proof:** "Is the same form as" is an equivalence relation because each simple path is mapped individually and unambiguously into a path form. Hence, each simple path is in some path. These paths are either identical or distinct so the relation satisfies the reflexive, symmetric, and transitive properties.
To show that this equivalence relation imposes a partition on the extension we can assume without loss of generality that the rule is defined on its entire domain. Then, each S-B pair is associated with some simple path of the rule and, in turn, with some path. To show that the equivalence relation is mutually exclusive (i.e., that the classes of S-B pairs associated with the paths of the rule are disjoint), we need only note that there is one and only one simple path, and hence path, associated with each S-B pair.

**Theorem B:** Each rule has a finite number of different paths.

**Proof:** The observation is obviously true of rules with a finite number of simple paths. Suppose the rule has an infinite number of simple paths. Then, all but a finite number of them must include at least one repetition (i.e., one repeated subsequence) because each rule contains at most a finite number of atomic or's. We need to show that there can be at most a finite number of different repetitions. But, this follows directly since there are only a finite number of subsequences which can be repeated.

**Corollary A:** The paths associated with a rule partition its extension into a finite number of equivalence classes.

**Corollary B:** The paths of each subrule of a rule impose a finite partition on the extension of the subrule.

As an exercise, consider the rule

\[
\begin{align*}
{\text{Start}} & \xrightarrow{r_1} \quad {\text{Halt}} \\
\quad & \xrightarrow{r_2}
\end{align*}
\]

where \( r_1 = xxBy \rightarrow xB0y \) and \( r_2 = xxabaBy \rightarrow xBly \) (\( x \) is a string of a's and \( y \) is a string of 0's and 1's). Identify a computation, a simple path, and a path of this rule. Also identify the partition imposed by the paths of this rule on its extension.

In discussing the problem of assessing behavior potential we use the following terminology.

**Definition 10:** A goal situation \( (S, G) \) is said to be resolved iff some behavior \( B \) is observed which either satisfies \( (S, G) \) or does not satisfy it. We say that a subject achieves the goal of \( (S, G) \) if he
generates a behavior which satisfies \((S_o, G)\). In this case, he is also said to succeed. He fails if he does not succeed.

In \(\mathcal{K}\) the subject is theoretically given all of the time he needs to respond. Practically speaking, however, to allow for situations where the computation may not halt, we must impose some time criterion.

**Definition 11:** A trial consists of a goal situation \((S_o, G)\) and its resolution. An assessment episode is a trial in which \((S_o, G)\) is satisfied by an \(r\) in \(K\). An assessment program (criterion referenced test) is a finite sequence of assessment episodes, one for each path of each rule in \(K\). A learning episode is a trial in which \((S_o, G)\) is not satisfied by an \(r\) in \(K\). A training program consists of a finite sequence of learning episodes.

**Definition 12:** A goal \((S_o, G)\) is satisfied by a rule \(r\) iff \(r\) \((S_o)\) satisfies \((S_o, G)\). We also say that \((S_o, G)\) is satisfied by the particular path of \(r\) associated with \((S_o, G)\) where the output \(B_o\) of the computation associated with \(S_o\) and \(r\) satisfies \((S_o, G)\). A subject \(S\) is said to know a path iff he can perform perfectly in all goal situations satisfied by that path. The rule corresponding to the set of known paths of a rule \(r\) is denoted \(r_s\) (for subject's rule). The set of all known rules in \(K\) is denoted \(K_s\).

Given any computable function (i.e., set of input-output pairs generable by some rule), it is well-known that there is a countably infinite number of rules which might account for it. In practice, of course, the number of such rules that might reasonably be employed by a human subject is typically quite small (and often just one). Subtraction, for example, is typically performed via one of two methods, borrowing or equal additions.

In general, it is possible to distinguish two or more rules extensionally (i.e., in terms of observables) just to the extent that they impose different partitions on their common extension. Hence, for purposes of prediction, we identify all rules imposing identical partitions. That is, we do not discriminate among such rules since for purposes of explanation and prediction we cannot distinguish among them. Indeed, because we shall eventually want to allow rules to act on rules, we identify all subrules with identical partitions. (We do not attempt a formal justification for this statement but simply note that in operating on rules a rule may act on subrules.)

**Definition 13:** Two or more rules (subrules) are said to be partitionally equivalent (equivalent) iff they impose identical partitions on
a common extension.

Where alternative rules (e.g., borrowing and equal additions—see Durnin & Scandura, 1971) impose different partitions on an extension, the situation is somewhat different. Assuming (as we do below) that paths of a rule are either totally available or completely unavailable, then different patterns of behavior (i.e., successes and failures) on a given extension may more directly reflect one of the rules than the others. The pattern of successes and failures in subtraction, for example, depends to a great extent on whether borrowing or equal additions is used. Furthermore, it is reasonable to expect that some subjects may use a combination of two or more such rules.

For this reason, whenever it seems likely that two or more nonequivalent rules (with a common extension) may be used in generating behavior, we replace the nonequivalent rules with a single rule that is equivalent to them. To see how such a rule may be devised it is sufficient to note two things: (1) A set of $n$ nonequivalent rules with a common extension imposes a refined partition on the extension which consists of all $n$-fold intersections of equivalence classes involving one equivalence class from each of the $n$ partitions associated with the $n$ nonequivalent rules. More exactly, if $\{A_{11}, A_{12}, \ldots, A_{1m_1}\}, \{A_{21}, A_{22}, \ldots, A_{2m_2}\}, \ldots, \{A_{n1}, A_{n2}, \ldots, A_{nm_n}\}$ are partitions imposed by nonequivalent rules $r_1, r_2, \ldots, r_n$, then the refined partition is $\{(A_{i1} \cap A_{i2} \cap \ldots \cap A_{in_n}) \mid 1 \leq i_1 \leq m_1, 1 \leq i_2 \leq m_2, \ldots, 1 \leq i_n \leq m_n\}$. (2) It is always possible to devise a new rule which imposes the refined partition on the common extension. Furthermore, any other rule which imposes this refined partition is equivalent.

From now on we shall assume that the extensions associated with the rules in $K$ are disjoint and further that no two rules have the same extension. In addition, to insure that $K$ provides an adequate basis for $\mathcal{F}$, it is implicitly assumed that all rules known to $S$ that are relevant to $\mathcal{P}$ are embedded in the rules in $K$. It is also assumed that the rules in $K$ are refined to the point where each atomic or and any acts in an all-or-none fashion. (This is always possible in principle since if I interpret the notion properly every Turing machine has this property.)

The following two axioms provide a basis for assessing behavior potential (i.e., finding out which paths of which rules in $K$ the $S$ knows).

**Axiom 1:** There is exactly one nonempty criterion referenced test and one training program.
**Axiom 2:** If a subject achieves \((S_o,G)\) that can be satisfied by a path of a rule \(r \in K\) (there is exactly one such rule), then he knows the path (i.e., can achieve all \((S_o,G)'s\) for that path); if he fails on one such \((S_o,G)\), then he will fail on all.

**Theorem (Assessing Behavior Potential):** Given a subject \(S\) and a class of \(S-B\) pairs which can be accounted for by an \(r \in K\), then there is a finite number of \((S_o,G)'s\) which can be used to determine which paths of \(r\) the \(S\) knows (i.e., to determine the corresponding \(r_s\)).

**Proof:** We prove the theorem by showing how to select the \((S_o,G)'s\). Each \(r \in K\) has a finite number of paths and thus partitions the given class (of \(S-B\) pairs). Select one \(S_o-B_o\) pair from each equivalence class in this partition. Taking Axiom 2 into account, the \((S_o,G)'s\) which correspond to these \(S_o-B_o\) pairs can be used as follows to determine which paths of \(r\) are known: If the subject achieves an \((S_o,G)\) satisfied by a path of \(r\), then he knows that path; otherwise he does not.

This theorem provides a basis for assessing a subject's behavior potential (relative to a given \(K\)).

**Corollary A:** Given any \(K\), a finite number of \((S_o,G)'s\) is sufficient for determining \(K_s\).

**Proof:** This follows because the number of rules in \(K\) is finite.

**Corollary B:** If all rules in \(K\) are atomic, then just one \((S_o,G)\) is needed for each rule in \(K\).

**Proof:** This follows because atomic rules have only one path.

A. As an exercise, consider the rule \(r\) above together with the related rule \(r'\)

\[
\begin{array}{c}
\text{Start} \\
\downarrow r_1' \\
\text{r_2'} \\
\downarrow \\
\text{Halt} \\
\end{array}
\]

where \(r_1' = \text{xxxxBy} \rightarrow \text{xB00y}\)
\(r_2' = \text{xxxxaBy} \rightarrow \text{xB0ly}\)
\(r_3' = \text{xxxxaaBy} \rightarrow \text{xB10y}\)
\(r_4' = \text{xxxxaaaBy} \rightarrow \text{xB1ly}\)

\(\text{dom}' = \{\text{xxxxBy}, \{\text{xxxxaBy}\}, \{\text{xxxxaaBy}\}, \{\text{xxxxaaaBy}\}, \{\text{By}\}\} \).
Also let the common extension be \( \{(wB, Bz) \mid w \text{ is a string of a's and } z \text{ is the binary numeral representing the number of a's}\} \). Identify a single rule which is partitionally equivalent to \( r \) and \( r' \) collectively.

B. Suppose \( r \) is in \( K \) and that \( S \) is tested and responds as indicated below.

\[
\begin{align*}
\text{aaaB} & \rightarrow 11 \\
\text{aaB} & \rightarrow 1
\end{align*}
\]

Use the axioms to identify which expressions (of the form \( wB \), \( w \) a string of a's) \( S \) may be expected to respond to successfully and which not.

C. The behavioral reality of this approach is demonstrated in Section 3 of Chapter 7. See Durnin and Scandura (1971) for a more complete discussion of the experimental results and comparison of the algorithmic approach formulated here with item forms and hierarchical technologies.

The assessment procedure described above clearly provides an explicit basis for determining \( K_s \). Our second major step is making precise the mechanisms by which the rules in \( K_s \) are put to use, and new rules are acquired (i.e., \( S \) learns).

We begin by defining the notion of higher order goal relative to a given goal situation. Consider goal situation \( (S_o, G) = \{R \mid P(S_o, R) \} \) where \( P \) is the predicate of \( G \). Then

\[
\text{Definition 14: } \text{The second order goal relative to } (S_o, G) \text{ is}
\]

\[
G^2 = \{r \mid S_o \in \text{Dom } r, \text{Ran } r \subseteq G\}
\]

Similarly,

\[
G^3 = \{r \mid S_o \text{ together with the } r's \in K_s \text{ specify an element } \\
\in \text{Dom } r, \text{Ran } r \subseteq G^2\}
\]

\[
\vdots
\]

\[
G^n = \{r \mid S_o \text{ together with the } r's \in K_s \text{ specify an element } \\
\in \text{Dom } r, \text{Ran } r \subseteq G^{n-1}\}
\]

Notice that the elements in the domain of any \( r \) in a goal of third or higher order consist (partly) of rules in \( K_s \). Also notice that rules which satisfy higher order goals need not necessarily satisfy \( (S_o, G) \).

In particular, it may be that \( S_o \in \text{Dom } r \) and \( \text{Ran } r \subseteq G \), but \( R_o = r(S_o) \) may not be in \( (S_o, G) = \{R \mid P(S_o, R)\} \). As we shall see below, our definition allows for "false starts" in problem solving (cf. Axioms).

\[
\text{Definition 15: } \text{The } m\text{th stage of a trial is a triple } \langle(S_o, G), G', K_s^m \rangle \text{ where } (S_o, G) \text{ is the goal situation of the trial, } G' \text{ is the } m\text{'th order goal relative to } (S_o, G) \text{ where } m' \leq m, \text{ and } K_s^m \text{ is the set of }
\]
rules known to the subject (at the $m$th stage). The $m'$th order goal is said to be in control during the $m$th stage of a trial.

The following axioms spell out the conditions governing learning and performance under memory-free conditions.

**Axiom 3:** At stage $\langle S_0, G \rangle, G^m, K^m$ if there is exactly one $r \in K^m_S$ that satisfies $G^m$, then control goes to $G^{m-1}$ (at the $m+1$st stage) where $m' \geq 2$ and $r$ is applied. Where $m' = 1$, the computation Halts. Otherwise (i.e., if $G^m$ contains no rule in $K^m_S$ or more than one), control goes to $G^{m+1}$ (i.e., the next stage is $\langle S_0, G, G^{m+1}, K^{m+1}_S \rangle$) if $G^{m+1}$ exists. Where $G^{m+1}$ does not exist, the computation halts (unless there is more than one rule in $K^m_S$ in which case the selection is nondeterministic).

Recall that the existence of $G^m$ for a given $m$ depends on the availability of a suitable $\text{dmc}$. $\text{dmc}$'s unfortunately do not reside in $K^m_S$ as such (where $K^m_S$ is the set of rules available to $S$ at the beginning of the trial). They are nonoperational in the sense that their presence cannot be determined directly. The best our assessment procedure can do is to specify a "decision" rule consisting of the $\text{dmc}$ in question together with $\text{or}$'s which attach common responses to stimuli in each equivalence class of the partition ($\text{dmc}$).

**Axiom 4:** An output rule $r$ generated at any stage automatically becomes part of the available knowledge at the next stage (i.e., $K^{m+1}_S = K^m_S \cup \{r\}$).

If we denote the knowledge had by $S$ at the beginning of trial $n$ by $K^m_{S_n}$, then $K^m_{S_n}$ is the knowledge had by the subject during the $m$th stage of the $n$th trial. Notice that any finite number of rules may be added to $K^m_{S_n}$ on any given trial.

**Axiom 5:** If the output $r$ generated at any stage satisfies $G^m$ (irrespective of how many rules in $K^m_S$ satisfy $G^m$), then control reverts to $G^{m-1}$ where $m' \geq 2$ or STOP where $m' = 1$ and $r$ is applied. Otherwise, the last rule (in $K^m_S$) selected in the course of the

---

4Because no operations are performed, $\text{dmc}$'s themselves may be assumed to take place instantaneously—or at least in some short fixed time. Of course, the operations which must follow if a $\text{dmc}$ is to be detected do result in a measurable response latency.

Latency measures may provide one method for determining the internal structure of rules. (For one approach to this problem, as well as a concise summary of its long history, see Sternberg, 1969. Also see Chapter 8 for a related, detailed discussion of information processing.)
derivation is (temporarily) eliminated (from \( K_{S}^{m} \)) and control at the next stage shifts to \( G^{m'} \) where \( n' \) is the level in the preceding derivation at which the eliminated rule was selected.

The reason for eliminating rules (temporarily) from \( K \) is to allow for "false starts" and subsequent attempts at problem solving. Some such mechanism seems essential if we are to reflect human behavior.

Although the above definitions and axioms (in one form or another) may be expected to play a central role in the idealized theory, they alone will probably not be sufficient. In any complete theory, for example, it will be necessary to make explicit the sense in which the (idealized) theory of learning depends on the underlying competence theory. Definition 16 together with Axiom 6 provide one way of accomplishing this.

**Definition 16:**

(a) Given a goal situation \( \langle S_{o}, G \rangle \), \( G \) is **reachable** from stimulus \( S_{o} \) via a subset \( A \subseteq K \) if there is an \( R_{o} \in \mathcal{R} \) which satisfies \( G \) such that \( S_{o} \Rightarrow R_{o} \) is \( n \)th order generable from \( A \) (i.e., such that there is an \( r \in A^{n} \) (for some \( n \)) which accounts for \( S_{o} \Rightarrow R_{o} \). (In this case, we also say that \( \langle S_{o}, G \rangle \) can be *satisfied* by a subset \( A \subseteq K \).)

(b) A rule \( r \in K \) is **relevant** to \( \langle S_{o}, G \rangle \), if there is a subset \( A \subseteq K \) with \( r \in A \) such that \( \langle S_{o}, G \rangle \) can be satisfied by \( A \) but not by \( A - \{r\} \).

(c) The set of rules \( K_{rel} \langle S_{o}, G \rangle = \{ r \in K \mid r \text{ is relevant to } \langle S_{o}, G \rangle \} \) relevant to \( \langle S_{o}, G \rangle \) is called the **relevance class of** \( \langle S_{o}, G \rangle \).

(d) The relevance class of a training program, \( K_{rel} \), is the union of the relevance classes of the \( \langle S_{o}, G \rangle 's \) in the training program.

**Axiom 6:** Let \( K_{rel} \) be the relevance class of the training program of the theory \( \mathcal{L} \) relative to the set \( K_{S} \) of rules actually known to \( S \). Then, \( K_{rel} \subseteq K \) where \( K \) is the rule set (in \( \mathcal{K} \)) introduced by the competence theorist.

Axiom 6 says, in effect, that the theory \( \mathcal{L} \) can be no better than the theory \( \mathcal{K} \) on which it is based.

Another assumption that needs to be made explicit concerns the extent to which an experimenter may control behavior by manipulating goals. In particular, it is necessary to assume that presenting a goal situation invariably results in \( S \) adopting the goal. Otherwise, the theory would lose predictive value and be useful only for explanation. More precisely, we assume not only that \( S \) is able to interpret any goal situation description presented by the experimenter but also, that he will do so. Furthermore, once understood this goal is assumed to be in control at the first
stage of the trial.

We shall not attempt to develop these fine points or to draw out further implications\(^5\) of the theory. This will be the subject of future papers.

3. APPLICATIONS

(1) Let \( K_s = \{ r_1, r'_1, r_2, r'_2, \ldots, r_n, r'_n, o, i, d \} \) where \( r_i \) and \( r'_i \) are rules (with disjoint extensions) such that \( \text{Ran } r_i \subseteq \text{Dom } r'_i \) for \( i = 1, 2, \ldots, n \), \( o \) is composition, \( i \) is identity, and \( d \) is a decision rule for determining for any given \( r \) and \( (S, G) \) whether or not \( S \in \text{Dom } r \) and \( \text{Ran } r \subseteq G \). Also let \( (S_o, G) = \{ R \mid P(S_o, R) \} \) where \( S_o \in \text{Dom } r_3 \) and \( \text{Ran } r'_3 \subseteq G \). Then, \( G^2 = \{ r \mid S_o \in \text{Dom } r, \text{Ran } r \subseteq G \} \). The composite \( r_3 \circ r'_3 \) satisfies \( G^2 \) and can be generated by applying \( o \) to \( (r_3, r'_3) \). Hence, \( (S_o, G) \) can be satisfied and \( S \) will learn the rule \( r_3 \circ r'_3 \) in the process.

Now let \( (S'_o, G) \) be such that \( S'_o \in \text{Dom } r_{n+1} \) and \( \text{Ran } r'_{n+1} \subseteq G \) where \( r_{n+1} \) and \( r'_{n+1} \notin K_s \). Can \( S \) satisfy \( (S'_o, G) \)? How could \( K_s \) be modified so that this would be possible?

(2) Semantics can be brought into the picture by modifying the example above so that \( p_i \) is a program for \( r_i \) and \( m \) is a rule for assigning meanings to arbitrary programs of the given types. Let \( K'_s = \{ m, o, \text{conj}, i \} \) where \( \text{conj} \) is conjunction. Furthermore, we allow programs \( p_i \) in the goal environment \( (S_o, G), E \) where \( p_i \in E - S_o \). In this case, even a goal situation that can be satisfied by one of the rules \( r_i \in K_s \) (above), strictly speaking, requires a derivation. Here, the meaning rule \( m \) is applied to \( p_i \) in the environment to generate \( r_i \) which satisfies \( G^2 \). (For \( G^2 \) to be meaningful, of course, we would need to include a suitable decision rule in \( K'_s \).)

A third level derivation is required where a given goal situation can be satisfied only by a composition \( r_i \circ r'_i \). To derive \( r_i \circ r'_i \) we need a conjunction (\( \text{conj} \)) of \( m \) followed by composition (\( o \)), applied to \( p_i, p'_i \). But to get this we need to apply "\( \text{conj} \circ \text{composition} \)" to \( m \) and \( o \). This latter rule results on applying "\( o \)" to \( \text{conj} \) and \( o \).

The distinction between rules and programs may be suppressed in experiments where meaning is not at issue. Indeed, this was the case in all of the experiments reported in Chapters 7 and 8 and Chapter 7 of Volume II.

---

\(^5\)As this book goes to press, I am in the middle of a project on mathematical problem solving (NSF Grant #GW 6796) that among other things is aimed at drawing out implications of this theory for artificial intelligence and mathematics education. In the future, I also hope to draw out implications of the theory for optimizing learning.
Foundations of the Idealized (Memory-Free) Theory

The study on verbal and symbolic statements of rules reported in Chapter 2, Volume II involves meaning more or less directly.

The reader with a computer science orientation may wonder why we have made a distinction between rules and shifting goals since the role of the latter may be incorporated into single unified procedures (rules). Thus, in example (2) it would be a simple matter to devise a unified procedure that would encode and operate on programs in the environment to produce new procedures (rules)—and, then, would "turn around" and apply these newly formed procedures to that portion of the environment that we have called simple stimuli. In effect, the interpretation and application of programs \( p_i \) which we have explained above in terms of shifting goals, could also be viewed in terms of applying a single procedure to the environment.

It would appear, therefore, that these two views are mathematically equivalent. But, they are not equivalent psychologically. The present view not only allows for a greater variety of behavior (because the higher and lower rules may act separately, for example, as well as in combination), but, more importantly, also provides theoretical constraints on rule based accounts which are directly reflected in human behavior (cf. Chapters 7 and 8). To the extent that such constraints are ignored, each behavior theory reduces to a rule; that is, the rule becomes the theory.

(3) Perceptual learning can be illustrated by extending the example on p. 110 as follows. Let \( K' \subseteq K_s \) where \( \Theta \) "forms the intersection." In this case, the goal \( G \) in \( \langle S_o, G \rangle \) consists of pairs of properties of stimuli (e.g., size and shape). Then, assuming \( K_s \) contains a decision rule for \( G^2 = \{ r | S_o \in \text{Dom } r, \text{Ran } r \subseteq G \} \), we see that \( S \) can satisfy \( \langle S_o, G \rangle \) while learning an \( r \in G^2 \).

(4) We can illustrate the "false start" phenomenon with the rule set \( K_s = \{ r_1, r_1', \ldots, r_n, r_n', c, h, s \} \) where \( r_1, r_1', \ldots, r_n, r_n' \) are as in (1), \( c \) is simple composition, \( h \) is a h.o. rule in which a simple composition is followed by a simplification (elimination subrule) which makes it possible to respond faster, and \( s \) is a selection rule which says simply select \( c \), if available, otherwise select \( h \). If \( S \) is posed with the simple task of solving a problem requiring a composite, say \( r_1 \circ r_1', \) then control will go first to \( G^2 \), and then to \( G^3 \). But, because there are two rules, \( 6 \)

---

6The outputs of \( c \) and \( h \) are behaviorally distinct because of the way we have defined rules in \( K \). Nonetheless, these outputs (rules) may have equivalent effects on elements in their respective domains with respect to goals which are "sufficiently" indiscriminate.
and \( h \), which satisfy \( G^3 \), control goes to \( G^4 \). Here, only \( s \) applies and \( o \) is selected. Control shifts downward and \( r_1 \circ r'_1 \) results. Finally, \( r_1 \circ r'_1 \) is applied to solve the problem. Now consider what happens if we impose a more stringent criterion. Not only must \( S \) get the correct answer but the answer must be generated quickly. In this case, the rule \( r_1 \circ r'_1 \) selected by \( o \) will not do; so \( o \) is eliminated and control goes to \( G^3 \).

Now \( h \) is the only rule which satisfies \( G^3 \) so control goes to \( G^2 \) and \( h \) is applied. Application of \( h \) yields a more efficient version of \( r_1 \circ r'_1 \), a version that when applied satisfies the initial goal condition.

4. EPILOGUE

The distinction between knowledge divorced from behavior, and knowledge attributable to particular subjects, is basic. The former view of knowledge corresponds directly to the notion of competence as used in generative linguistics. The latter view is more consistent with those of epistemologically oriented psychologists like Piaget where knowledge is a constructive potential for behavior. In this case too knowledge is assumed to grow according to postulated learning mechanisms.

What the present theory does (among other things) is to provide a missing link between the two views. Knowledge, viewed as competence, corresponds to a theory of the observer, and provides measuring units (rules) against which the knowledge had by individual subjects may be measured. In effect, the theory of knowledge (competence) provides a basis for the operational definition of human knowledge.

In retrospect, these two uses of the term "knowledge," and their relationship to existing theoretical approaches, might better have been emphasized by reserving the term "competence" for the former usage and the term "knowledge" for the latter.
So far, we have pretty much avoided any mention of memory in our formulation. The proposed learning and performance mechanisms have all assumed an information processor with perfect memory for previously acquired knowledge and with an essentially unlimited capacity for processing information. This definitely does not imply that our theorizing so far is of little value. That conclusion would be wrong on at least two counts. First, there are many practical situations in structural learning where memory is of minimal concern. In problem solving, for example, the subject is almost always given all of the paper, pencil and other memory aids (even text books) that he needs. Typically, we also do our best to insure that the necessary lower order rules are readily available. The concern is generally with whether or not the individual can integrate available knowledge to solve problems. Considerations such as whether he can do it in his head or not, time to solution, and so on are of secondary concern. Second, questions of memory can be eliminated in experimentation by insuring that the relevant procedures and memory aids are available to the subject. In order to empirically test the problem solving hypothesis of Section 4, Chapter 7, for example, we would first make sure that the subject can use all of the necessary higher and lower order rules in the kinds of situations required before testing him on a critical test problem.

The most important value of memory-free theorizing, however, may be that it provides a basis from which other theorizing might begin. This entire monograph, in fact, is based on the contention that any complete account of complex human learning and behavior will necessarily involve theorizing at at least three independent but complementary levels, each with its own kind of empiricism.

The aim of this chapter is to show in outline form how the idealized
theory may be extended to include memory. We begin with an introduction
to the problem of memory and some background, and then consider some general
similarities and differences between existing memory theories and the type
proposed here. In Section 2, we outline a new theory of memory which builds
on our previous work. Since the treatment is self-contained, the reader
who is not interested in the general background may turn directly to this
section.

1. NATURE OF THE PROBLEM

There are two major problems with which any adequate theory of memory
must deal. The first concerns the way in which (a) incoming information
in the form of stimulation from the outside world is stored in (long-term)
memory, (b) stored information may be modified over time through learning,
and (c) stored information is retrieved. Two of my own studies (Scandura,
1967; Scandura & Roughead, 1967), for example, were concerned with the pro-
cesses involved in (a) recoding (storing) lists of nouns according to common
concept categories, (b) determining the influence of different kinds of
interpolated activity on what was stored, and (c) retrieving the lists
during recall.

The second problem concerns the processing of information and, particu-
larly, the limited amount of information which human beings can process at
any given time (e.g. Miller, 1956; Chomsky & Miller, 1963). In this case,
one might be concerned, for example, with accounting for the differential
ability of human subjects to perform mental arithmetic with large and small
numbers even where the subject knows perfectly well how to compute. Thus,
if given a pair of numbers like 23 and 12, most adults would have little
difficulty in (mentally) generating the sum, 35. With a pair like 478,947
and 52,762,894, on the other hand, most would not fare nearly so well.

Unfortunately, it has been difficult to separate these problems in
empirical studies. Much of the more recent work, for example, has been
influenced by a new technique for measuring retention which was introduced
by Peterson and Peterson (1959). The basic idea of their experiment was to:
(1) present CVC nonsense syllables, (2) have the subject count backward by
threes or fours, and (3) test him, after some intervening period ranging
from about 0 to 30 seconds, to see if he could remember the given nonsense
syllable. Contrary to the then prevailing expectation of most psychologists,
they found that retention decreased rapidly over this period. This led to
a good deal of subsequent research and the basic paradigm is still in wide
use today.
The problem with this type of study is that it does not distinguish operationally between mechanisms associated with the storage, modification, and retrieval of information, on the one hand, and the limited ability of the human subject to process information, on the other. Thus, in a Peterson and Peterson type situation, a subject may attempt to retain a nonsense syllable either by continuing to process the information (by a process typically referred to as rehearsal), or by storing it in (long-term) memory. Under these circumstances, it is difficult to say anything definitive about either type of mechanism as a result of the experimental data obtained.

1.1 Current Approaches to Memory Theory

Whatever its limitations, the Peterson and Peterson study helped to initiate a revival of interest in memory. A large amount of data has been collected during the past decade or so and memory theorists have taken up the challenge of accounting for it (e.g., Norman, 1970; Bower, 1967; Atkinson & Shiffrin, 1968).

In large part because of the influence of computer science and transformational linguistics, memory has been approached more and more as part of an information processing system. According to Norman (1970, p. 2), the general picture is this.

"First, newly presented information would appear to be transformed by the sensory system into its physiological representation..., and this representation is stored briefly in a sensory information storage system. Following this sensory storage, the presented material is identified and encoded into a new format and retained temporarily in a different storage system, usually called short-term memory. Then, if extra attention is paid to the material, or it is rehearsed frequently enough, or if it gets properly organized, the information is transferred to a more permanent memory system (or, in some models, the rate at which it decays decreases substantially). In general, the capacity of this more permanent storage is so large that information that is stored there must be organized in an efficient manner if it is ever to be retrieved. Then, finally, when it is necessary to retrieve information from memory, decision rules must be used, both to decide exactly how to get access to the desired information and then to decide exactly what response should be made to the information that is to be retrieved."

Norman (1970, p. 2) goes on to point out that although different theorists have tended to emphasize different aspects of this system, there is fairly uniform agreement on the major essentials. Most agree that there
are three different types of storage systems: a sensory information storage, a short-term memory, and a long-term memory. Theorists with a more traditional orientation (e.g., Melton, 1963; Postman, 1964), nonetheless, have tended to argue that retention over short periods of time (e.g., up to one minute) is subject to the same influences as retention over longer periods and hence that there is no need to distinguish between short and long-term memory. However, in order to deal with the confounding influence at recall of information currently being processed, traditional theories of (long-term) memory have been modified and elaborated to such an extent that one wonders whether or not they are really the same type of theory after all. For a recent and well formulated theory along these lines, see Bernbach (1970).

The others feel that the distinction between short and long-term storage systems is absolutely necessary. (Wickelgren (1970), in fact, postulates a fourth system which operates between short and long-term memory.) This alternative view is well exemplified by the recent work of Atkinson and Shiffrin (1968). We outline the major features of this theory in some detail because it seems sufficiently close to our own theorizing to provide an instructive basis for comparison.

Atkinson and Shiffrin (1968) make a basic distinction between permanent, or what they call structural, features of memory which remain invariant, and control processes which are under the voluntary control of the subject.

The permanent or structural features of memory may be summarized as follows. All incoming information enters the organism through the sensory register. Some of this information is lost, and the rest goes into short-term store. It is assumed that incoming stimulation is stored for a period of up to several hundred milliseconds in the sensory register (which may be thought of as the retina of the eye for visual stimulation) before it enters short-term store. For present purposes, questions related to the sensory register can safely be ignored and are discussed here only incidentally.¹

Information may also be lost from short-term store. Unless rehearsed,

¹ While important in dealing with extremely short-term memory phenomena of the sort investigated by Sperling (1960, 1963), Averbach and Coriell (1961), and Estes and Taylor (1964, 1966) among others, the physiological processes involved in "copying" incoming stimulation on the sensory register are seldom of central, or even peripheral, concern in structural learning. In learning mathematics, for example, we rarely (if ever) are concerned with what might happen, say, if a formula is flashed on a screen for a brief period of 100 milliseconds followed immediately, say, by a field of bright polkadots. In typical situations, we can safely ignore extreme conditions of this sort and concern ourselves exclusively with questions of long-term memory and the processing of information.
this information is assumed to decay with a half-life of between 10 and 15 seconds. The rest either remains in short-term store (where it is rehearsed), or it is transferred to long-term store. Information in long-term store, in turn, may be transferred back to short-term store. Long-term store is assumed to be permanent, but may be modified due to subsequent learning. Learning mechanisms, however, are not discussed in any detail. These structural features impose certain restrictions on the way memory may operate. In the Atkinson and Shiffrin system it is not possible, for example, to transfer information to long-term store directly from the sensory register without first going through short-term store.

The processes by which all this is accomplished are assumed to be under the voluntary control of the subject. The subject is assumed to have available a variety of control processes by which he can copy information at one level onto some other level (allowed by the structural features of memory). In particular, information may enter short-term store both from the environment (i.e., sensory register) and from long-term store. There are assumed to be three kinds of processes within short-term store: (1) processes for searching short-term store and for retrieving information within it, (2) rehearsal processes by which information in short-term store may be kept from "decaying," and (3) processes for transferring information from short-term store to long-term store. There are also assumed to be specific search and retrieval processes associated with long-term store. It is assumed that a subject may fail to retrieve information in long-term store either because the control (i.e., search) process he uses is inadequate and/or because the way in which the information was stored has been modified.

1.2 Relation of the Atkinson-Shiffrin Theory to the Present Approach

The Atkinson and Shiffrin theory clearly deals with most of the problems which are assumed to be critical to any adequate theory of memory. The theory also has certain things in common with the idealized theory of Chapters 7 - 9. Most obvious is the fact that both theories involve information processing. In particular, many of the control processes in the Atkinson and Shiffrin theory have direct counterparts in our own theorizing. Rehearsal, which is the only control process they identified which both begins and ends in short-term store, seems to be but one of an indefinite number of procedures which we have assumed are routinely carried out by human beings. In this sense, rehearsal is no different from applying an algorithm to add numbers or applying a truth table argument to determine the truth or falsity of a statement in the
propositional logic.

In spite of such similarities, the differences are more fundamental. For one thing, Atkinson and Shiffrin appear to attribute somewhat more physical reality to their permanent features of memory than has been the case in our theorizing. A result of this commitment to physical structures has been a failure, I think, to clearly separate theoretical constructs according to function. Thus, questions relating to the processing of information and questions relating to the storage, modification, and retrieval of information overlap in the Atkinson and Shiffrin system in sometimes confusing ways.

They have proposed, for example, that information processing takes place in long as well as short-term store. Thus, the retrieval of information from long-term store is assumed to be a control process operating in long-term store. They have also had to deal with such anomalies as storing and retrieving information in short-term store.

In my own theorizing, I have been more inclined to make a sharp distinction between rules or processes (i.e., knowledge), on the one hand, and information processing or the use of rules and processes, on the other. In this sense, processes for storing and retrieving information, and rules for generating responses or other rules, or for interpreting statements in a language, are fundamentally no different. They are all put to use by a common information processor (an information processor with a fixed finite capacity to process information). Furthermore, in the theory outlined in Section 2, there is only one place to store information, and that is (long-term) memory. Information which is being processed is available by definition and does not require special retrieval processes. (To think of "storing" information in a central processor seems rather awkward.) In short, there are only two constructs and each plays a specific role in the theory; no physical reality is implied.

As a result of this difference, Atkinson and Shiffrin have at their disposal a much larger variety of explanatory mechanisms than we do. Consider, for example, the task of accounting for a subject's failure to recall, say, after 15 seconds. Atkinson and Shiffrin might attribute such failure to inadequate retrieval processes at two (or more) distinct levels. We, on the other hand, would be limited to an inability to retrieve (via information processing) from (long-term) memory. In order to get around the fact that some things are easier to recall than others, I have, in some of my own research, proposed that information may be stored either in relatively stable categories, like "small," "large," or "round," or in rather transient and difficult-to-locate categories such
as those based on rhythm and other transitory bases for reorganization (Scandura, 1967; Scandura & Roughhead, 1967). Greeno (1967) has offered a similar interpretation in terms of good and bad codes in long-term memory. In this case, at least, there is some question as to whether the extra (short-term storage) mechanism is necessary.

Perhaps the most important difference between the Atkinson and Shiffrin theory and our own theorizing, however, is in the nature of explanation itself. Thus, in the former theory, the basis for both explanation and prediction rests with such assumptions as "information is transferred with probability $a$" and "information in short-term store decays or is lost with probability $b$." In our view, "decay" itself is a process to be explained and is not a basis for explanation.² As suggested above, for example, decay in short-term store might equally well be attributed to a breakdown in retrieval. That is, rather than undergoing processing as supposed, the information in question may be displaced by a new element which results in overloading the information processor, and effectively placed in long-term store in an inaccessible location.

While recognizing that uncertainty will always play a role in human behavior, I have argued consistently (or, at least, persistently) that it may be possible to deal with that uncertainty in quite a different way than has so far been the case in psychology, that is by deterministic theorizing relative to the adequacy of certain boundary conditions which determine the domain of applicability of the theory.

2. GENERAL NATURE OF THE THEORY OF MEMORY

In the idealized theory, it is assumed, essentially, that the subject has a single memory, consisting of entities (that are not rules), rules, and rules which operate on rules. Furthermore, the contents of this memory, including new elements which may be generated in the course of a computation, are assumed to be readily and uniformly available to the subject. As pointed out above, this idealization can be approached in practice, but it is clearly not the norm. At any given point in time, people are more attuned to certain information than other. Indeed, psychologists refer to an element of knowledge that is readily available as being "aroused." In addition, information that has been stored in memory generally varies from time to time in the ease with which it can be recalled. We have all

² A similar comment may also be made about "stimulus generalization." Rather than providing a basis for explaining transfer, stimulus generalization itself is something to be explained in terms of what is learned (cf. Scandura, 1968).
experienced the inability, for example, to recall some name or fact that ordinarily would be readily available—the "tip of the tongue" phenomena.

Fortunately, the idealized theory can be "enriched" in a rather natural way, so as to handle these general phenomena (although we shall only consider such phenomena in regard to structured knowledge). In this enriched theory, we distinguish between a memory ($M$) and that (small) part of it ($A$) which is active at any one time. $M$ is a (large) finite set of rules as before; but only rules and (encoded) stimuli in $A$ can generate responses or produce new knowledge. In effect, all processing goes on in $A$.

For purposes of exposition, it is convenient to distinguish between a memory theory where it is assumed that the capacity of $A$ (as well as $M$) is finite but unbounded, and where the capacity of $A$ is fixed. This distinction has important empirical implications as well. The conditions under which the former theory is applicable are relatively commonplace and represent an important extension of the memory-free paradigm described in Chapters 7 and 8. The latter (enriched) theory follows Miller (1956) in assuming that $A$ contains $7 + 2$ "chunks" of information, only here the chunks may be rules as well as elements on which rules operate. This theory (presumably) reflects still more about behavioral reality and is applicable under an even broader range of conditions. (The more detail a theory is to reflect, of course, the more difficult it is to construct one.)

In the next three sections, we shall elaborate on this theory, providing examples and empirical evidence where available. Section 2.1 is devoted to the basic mechanisms of the theory, under the overly generous assumption that the capacity of $A$ is unbounded. In Section 2.2, we introduce additional mechanisms that make it possible to operate within the constraints of a fixed capacity processor ($A$). This effectively increases the domain of applicability of the theory. Unfortunately, there is little data to report; and the main justification for the theory is on post hoc and philosophical grounds. Finally, in Section 2.3, we develop and test a new experimental paradigm that may ultimately be useful in testing the theory. Specifically, we develop and test an analytical technique for calculating the memory load attached to specific rules. Then, we briefly sketch how this technique might be extended to provide a more general paradigm for testing the theory.

In no way, however, should the theory which follows be considered a finished product. Rather, it should be viewed as an approximation to the

---

3In view of the relative lack of directly relevant data, one can reasonably ask on what basis such a theory might be defended. No matter how elegant a theory, it must eventually make contact with reality if
truth. It should also be emphasized that this theory is primarily concerned with the role of memory in learning structured material. It is not to be expected that this theory should provide a precise account of the large and varied literature on short and long-term memory which for the most part has been based on unstructured experimental tasks. In fact, the theory actually suggests that different kinds of memory experiments may be needed if we are ever to understand how meaningful (structured) materials are learned, remembered, and used.

2.1 Memory Theory with Unlimited Processing Capacity

The memory theory with unlimited processing capacity is applicable under a considerably broader range of conditions than the idealized (memory-free) theory. In particular, the theory applies in situations where certain rules are not immediately available in A, even though the subject may have previously learned and stored them in M. The only essential condition is that for purposes of experimental test the subject have an unlimited capacity for processing information. He might, for example, be given all the paper and pencil he needs "for figuring."

Oddly enough, these conditions come precariously close to those required in the memory-free paradigm. In that case, too, if one examines carefully the experiments reported in Chapters 7 and 8, it becomes apparent that the needed information is typically not immediately available in the

(continued) it is to be more than an exercise in speculation. Nonetheless, parsimony itself is an important factor to be considered. Even where other alternatives are available, it is often easier to generate predictions where the basic assumptions are fewer in number. For the same reason, a parsimonious theory is often easier to understand.

Perhaps most important, however, all other things being equal, like the range of applicability, a parsimonious theory has a better chance of being correct (i.e., of being a closer approximation to the truth). Generally speaking, the fewer basic mechanisms (hypotheses) involved in a theory, the more varied are the situations with which these mechanisms have to deal. Correspondingly, our confidence grows in a theoretical mechanism to the extent that it provides an adequate account of a variety of different kinds of phenomena.

It is relatively easy to come up with mechanisms which provide adequate explanations for particular kinds of phenomena. It is much harder to come up with mechanisms which are more broadly applicable. Much the same, of course, may be said for empirical support. It is support in a variety of different kinds of situation which attests to the power of a hypothesis or theory, not just quantity of support per se. (A reasonable amount of support is also important, of course, because this increases reliability.)

Further, by definition the theory is not concerned with extremely short-term memory phenomena which involve physiological characteristics of human sensing organs (e.g., the eye). The theory is limited in that sense. Its potential usefulness depends on the accuracy of the contention that such characteristics are of minimal concern in structural learning.
processor, but rather is somehow represented in the environment (e.g., on cards used to denote rules). It is implicitly assumed that the subject is able to locate and encode the needed information at will. The memory theory with unlimited capacity must, in addition, make explicit the mechanisms by which needed elements are retrieved from memory (M).

In this theory, stimulation from the environment that enters A automatically becomes part of M. This information (i.e., stimulation which has been perceived) remains available to the subject as long as it remains in A. However, it can only be retrieved at a later time if it has been stored (via rules) in relation to other information which can serve to cue it. Specifically, storing information involves constructing rules by which the to-be-remembered elements (information) may be generated from elements in M.

Retrieving information involves using active rules to generate observables from given observables and elements in A. In A, control shifts among goals as before. The relatively small number of rules in A, however, serves to keep the number of goal levels and rule selections required within bounds. Where needed rules cannot be derived from rules in A, or needed inputs are not active, control shifts to the goal of retrieving the needed rules from active elements (rules and stimuli). To generate an adequate retrieval rule, control may shift levels as before. Once the needed information is retrieved, control returns to the goal from which the search was initiated, and the process continues.

Retrieval. The basic mechanisms of the memory theory with unlimited processing capacity are a direct extension of those for the idealized (memory-free) theory. Nonetheless, the number of possible conditions under which this theory applies are quite varied, and full appreciation of their scope and exact nature is perhaps best seen through examples. New mechanisms are required only in the case of retrieval, so we consider that first.

Without loss of generality, let us assume throughout that $S_0$ is the stimulus situation and G the goal where the response $R_0 = r_n \circ r_n(S_0)$ satisfies the goal situation $(S_0, G)$. $r_n$ and $r_n$ are arbitrary rules in the domain of the higher order composition rule $\circ$. We assume also that the subject can interpret $(S_0, G)$ and that $(S_0, G)$ remains in control throughout. Finally, we assume that the subject has available and can determine whether or not an arbitrary output satisfies each needed higher order goal associated with G.

Type I: Suppose that the solution rule $r_n \circ r_n$ is either in A or is derivable from rules in A.
Here, the theory reduces to the idealized (memory-free) theory, and exactly the same mechanisms apply. Notice in particular that a special case of this type occurs where the to-be-recalled element is itself in A—for example, when the element is being rehearsed. In this case, the element satisfies G directly. No derivation is necessary.

**Type 2:** Assume that \( r_n \) is not in A but is in M. Furthermore, \( r_n \) is retrievable via some rule \( r \) in A. To be more specific, let \( A = \{ r \} \) where \( r \) applied to the environment (E) yields \( r_n \). (When we say "applied to the environment," we include the possibility that \( r \) may apply to the goal as well as to the stimulus and other environmental stimulation.)

In this case, control goes to \( G^2 \) as before. Because \( r \) (the only rule in A) is not a member of \( G^2 \), control goes to \( G^3 = \{ r_j | \text{Ran } r_j \subseteq G^2 \} \).

Now, \( r \) satisfies \( G^3 \) so control reverts to \( G^2 \) and \( r \) is applied, yielding \( r_n \). This output in turn satisfies \( G^2 \) so control reverts to \( G \) and \( r_n \) is applied to \( S_0 \) giving the solution \( R_0 \).

A concrete realization of this situation is obtained by letting \( \langle S_0, G \rangle \) be "Find a number N such that \( N = 45 + 61 \)," \( r_n \) be an addition algorithm, and \( r \) be a rule which pairs stimulus situations which include +, -, x, or \( \div \) signs, respectively, with corresponding algorithms (rules). We might imagine that \( r \) is active (i.e., in A) because the subject has been asked to solve a series of similar computation problems in the immediate past.

**Type 3:** Next, assume that there is no suitable retrieval rule directly in A, but that such a rule can be derived from rules in A. As before, \( r_n \) is assumed to be in M. Specifically, suppose \( A = \{ r', r'' \} \) where \( r''(r') = r \) and \( r(E) = r_n \). Then, given the goal situation \( \langle S_0, G \rangle \), control goes to \( G^2 \) as before. Because no rule in A satisfies \( G^2 \) (recall that \( r_n \) satisfies \( G^2 \)), control goes to \( G^3 \). Again, however, there is no rule in A which satisfies \( G^3 \) (\( r \) satisfies \( G^3 \)), so control goes to \( G^4 \).

Here \( r'' \) satisfies \( G^4 \); control reverts to \( G^3 \) and \( r'' \) is applied to \( r' \) giving \( r''(r') = r \). But \( r \) satisfies \( G^3 \) so control reverts to \( G^2 \) and \( r \) is applied (to \( \langle S_0, G \rangle \)) yielding \( r_n \), which in turn satisfies \( G^2 \). Hence, control reverts to \( \langle S_0, G \rangle \), and \( r_n \) is applied to \( S_0 \) giving the solution \( R_0 \).

To see what is required in constructing a concrete realization of this type, it is instructive to consider the illustration of Type 2. In this case, the only essential change would be that of identifying rules \( r'' \) and \( r' \) such that \( r''(r') = r \) is a rule whose extension consists of pairs whose first elements are stimuli which include +, -, x, or \( \div \) sign and whose second elements are corresponding computational algorithms. Although it is possible to concoct rules \( r'' \) and \( r' \) which satisfy this condition,
no feasible possibilities occur offhand to the author. For a somewhat more reasonable example, let \((S_o, G)\) be "Find the number of inches I such that \(I=5\) yards," \(x\) yds. \(\rightarrow\) 36x in. be the solution rule, \(p\) generate solution rules from corresponding pairs of units (e.g., yds., in.), \(p'\) be "1 yd.=3 ft. and 1 ft.=12 in.," and \(p''\) operate on pairs of equivalences (e.g., \(p'\)) and construct rules like \(p\). \((p, p', p'')\) correspond to \(r, r', r''\), respectively.)

Although we assume that \(r'_n\) or \(r_n\) is in \(M\), it is worth noting at this point that it makes no essential difference whether or not a needed rule is in \(M\) as long as it can be retrieved (or derived) via rules in \(A\) alone. Furthermore, the mechanisms required can be identified (save for the distinction between \(M\) and \(A\)) with those used in the idealized (memory-free) theory.

Indeed, the distinction between retrieval and derivation rules is more a matter of convention than of substance. Both are higher order rules. Which term is used depends on whether the to-be-retrieved (derived) rule is assumed to be in \(M\) or must be derived for the first time. Nonetheless, retrieval rules generally are simpler and involve less complicated operations. Frequently, they are also less general. (See the discussion below for examples.)

The mechanisms of the idealized theory, nonetheless, are limited in an important sense. It is implicitly assumed throughout that all of the elements in the domains of rules involved in a derivation are immediately available. Although this is certainly a reasonable assumption to make in the idealized theory, it is not adequate where much of what has been learned in the past (in \(M\)) is not immediately available (in \(A\)). Types 4, 5, and 6 show how the above mechanism can be extended where elements in the domains of rules involved in a derivation must be retrieved or derived. Perhaps not so surprisingly in view of the above observations, \(M\) again plays a distinctly subordinate role in these mechanisms. Its sole justification, in fact, is to provide a cumulative record of the contents of \(A\). (The need for such a record becomes more apparent in Section 2.2.)

**Type 4:** Case 4 deals with situations where needed rules are generated by the action of rules initially in \(A\) on rules initially in \(M\). In particular, suppose that \(r'_n\) and \(r_n\) are in \(M\) and that \(o\) and a retrieval rule \(r\) for \(r'_n\) and \(r_n\) are in \(A\). That is, application of \(r\) to \((S_o, G)\) yields \(r'_n\) and \(r_n\).

In this case, control as before goes to \(G^2\): "Find a rule such that \(S_o\) is in its domain and whose range is contained in \(G\)." Because no rule in \(A\) satisfies \(G^2\), control goes to \(G^3\): "Find a rule whose range is contained in \(G^2\)." The (suitably restricted) composition rule \(o\) satisfies \(G^3\) so control goes to \(G^2\). But, \(o\) cannot be applied since nothing in its domain is in \(A\) (or in \(\langle S_o, G \rangle\)). In this case the new mechanism takes over and control goes to the goal \(G^3_{\text{Dom}(o)}\) of finding an element (i.e., a pair of
Memory Theory in Structural Learning 311  

rules) that is in the domain of \( o \). Because no element in \( A \) is in the domain of \( o \), control goes to the higher order domain goal \( G^{2}_{\text{Dom}2(o)} \) of finding a rule whose range is in the domain of \( o \). Now the rule \( r \) in \( A \) uniquely satisfies \( G^{2}_{\text{Dom}2(o)} \), so control reverts to the initial level \( G^{2}_{\text{Dom}(o)} \) and \( r \) is applied to \( \langle S_{o}, G \rangle \) yielding the pair \( \langle r^{n}_{n}, r_{n} \rangle \). This pair is in the domain of \( o \) (i.e., satisfies the initial domain goal) and, hence, control goes to the goal from which the diversion started, namely \( G^{2} \). At this point, \( o \) is applied to the pair \( \langle r^{n}_{n}, r_{n} \rangle \), which is now in \( A \), giving \( r^{n}_{n} \). The rule \( r^{n}_{n} \), in turn, satisfies \( G^{2} \) as before, so control goes to \( G \) and it is applied to \( S_{o} \) giving the solution \( R_{o} \).

The second example of Type 3 also serves here. We just modify \( p^{"} \) so that it generates the solution rule directly and add a rule \( r \) for retrieving \( p' \), the equivalences "1 yd. = 3 ft." and "1 ft. = 12 in." Without loss of generality we assume that \( r \) involves direct connections between "yards" and "inches" respectively, and the two equivalences. In this case, \( \langle r^{n}_{n}, r_{n} \rangle \) and \( o \) above correspond, respectively, to \( p' \) and \( p^{"} \). \( A \) initially contains \( p^{"} \) and \( r \).

**Type 5:** Type 5 is concerned with the reverse situation—where needed rules are generated by action of rules initially in \( M \) on rules initially in \( A \). For illustration, we assume that \( r^{n}_{n} \) and \( r_{n} \), together with retrieval rule \( r_{o} \) for \( c \), are in \( A \), but that \( o \) is in \( M \). Here, control goes to \( G^{2} \) and since no rules in \( A \) satisfy it, control goes up again to \( G^{3} \), and finally to \( G^{4} \). Now, \( r_{o} \) satisfies \( G^{4} \) so control goes to \( G^{3} \), \( r_{o} \) is applied and \( o \) is generated (in \( A \)). But, \( o \) satisfies \( G^{3} \) and control goes to \( G^{2} \) as before. Finally, \( r^{n}_{n} \) is generated, and the problem is solved.

Notice again in this case that the mechanisms of the idealized theory suffice without modification.

**Type 6:** Lastly, we consider a situation where needed rules are generated by the action of rules initially in \( M \) on rules initially in \( M \). Suppose that \( \langle r^{n}_{n}, r_{n} \rangle \) and \( o \) are in \( M \), but that retrieval rules \( r \) and \( r_{o} \), respectively, for \( \langle r^{n}_{n}, r_{n} \rangle \) and \( o \) are in \( A \). Then, control must go to \( G^{4} \) before some rule in \( A \) satisfies a goal. Here, \( r_{o} \) satisfies \( G^{4} \) so control goes to \( G^{3} \) and \( o \) is generated (in \( A \)). Rule \( o \), in turn, satisfies \( G^{3} \) and control goes to \( G^{2} \). But, \( o \) cannot be applied since no element in its domain is in \( A \) (i.e., no element satisfies \( G^{2}_{\text{Dom}(c)} \)). Hence, control goes to \( G^{2}_{\text{Dom}2(c)} \). Since \( r \) satisfies \( G^{2}_{\text{Dom}2(c)} \), control goes to \( G^{2}_{\text{Dom}(c)} \) and \( r \) is applied. The generated rules \( r^{n}_{n} \) and \( r_{n} \), then, satisfy \( G^{2}_{\text{Dom}(c)} \) (i.e., the pair is in the domain of \( c \)). Control, therefore, returns to \( G^{2} \), and \( o \) is applied to the pair \( \langle r^{n}_{n}, r_{n} \rangle \) giving \( r^{n}_{n} \) and the problem is solved.

These types are not intended to be exhaustive, of course, and any one situation involving retrieval may call for more than one type. Thus, for
example, one level in a derivation might involve type 6 and another type 4.

Storage. To complete our discussion, a few words about information storage are in order. In the present view, storing information involves forming a rule that can be used to generate the information. Specifically, the goal is to construct a rule whose extension includes an S-R pair consisting of some stimulus (which may be in the environment, in A, or in M) and the given information (as the response).

When looked at in this way, it is apparent that no new mechanisms are required for storage. In particular, the required rule may be in A—in which case nothing is required other than confirming that this possibility constitutes a special case of "storage," or it may be derivable from higher order storage rules in A (as above). The only thing new is that whereas the stimulus in retrieval is typically made explicit, the effective stimulus involved in storage is frequently (but not always) an unspecified element, usually in M. (What is "stored," however, is a rule, and not the response.) Storing information by forming rules corresponds in more traditional information processing views to inserting to-be-remembered items in particular locations (where the locations correspond roughly to rules). As we shall see below, ambiguity is typically introduced into memory research whenever the stimulus in question is not specified overtly.

Nature of Memory Research. In many studies of memory, each stimulus, as well as the goal involved, is made explicit during both storage and retrieval. During storage a rule is formed connecting the stimulus to the response. At retrieval the rule is applied to the given stimulus to generate that response.

Perhaps the simplest and most widely studied instance of this has to do with learning paired associates. In this case, both the stimulus and the response are presented to the subject and he is required to form an association (degenerate rule) connecting the two. Later, he is presented with the stimulus alone and is asked to give the response—in particular, to give a response which satisfies the frequently implicit goal of being equivalent to the previously appearing response.

There are many variants on this theme. We shall mention just one that recently has been studied rather intensively. In this case, the stimuli (e.g., dog) and responses (e.g., home) are embedded in simple sentences (e.g., the dog ran home). This method of presentation has had a reliably positive effect on recall. The reason for this result follows naturally when viewed within the present framework: The subject is not only given the stimulus and the response, but he is also given a simple rule which
connects them. In particular, the meaning of "The dog ran home" is stored as a unit so that when "dog" later appears as a stimulus, it is easy to generate "home." Subjects not presented with this connection must form their own; and frequently this is hard to do.

As one might suspect in view of the large amount of research in the area, there are many competing theories which deal reasonably well with paired associate learning. The main advantage of the present theory is that it applies equally well to more complex (structured) situations. Suppose, for example, that the subject is presented with a map of the United States showing the various state boundaries and the names of the states, and that he is asked to memorize them (the names), so that he can give them all back on presentation of the map sans names. The subject's task is to construct a rule which will allow him to give the names in the presence of the unlabeled map. To see that this may involve a bona fide rule, and not just a set of discrete associations, it is sufficient to note that subjects will typically develop different strategies for going through the states. Some, for example, might start with the New England states and work toward adjacent states systematically across the country. Others might start from the coasts and work inland; and so on. (This task came to mind while I was writing this because my son had just that morning been confronted with the task for homework.) It is also important to point out that the retrieval rules constructed during storage are the result of applying higher order rules. The former retrieval rule, for example, may be generated by application of a "work toward adjacent elements" higher order rule on pre-learned individual associations between state maps and names.

In other studies, only the storage and retrieval goals are given explicitly. The effectively operating stimuli are implicitly defined in terms of the rules (in A) used by the subject in achieving the goals. Consider, for example, recent work concerned with the rediscovered notion of mnemonics in memory. Here, one of the primary variables is whether or not the mnemonics, which act effectively as stimuli, are presented to the subject directly. In the preceding example, the map of the United States serves essentially as a mnemonic (stimulus) that may or may not be presented during storage or retrieval. It can only serve as a mnemonic, of course, if the map itself has been previously learned and is concurrently in A. During storage, presumably, the state names are embedded on the map. In any case, in recalling the names of the states, many of us quite consciously bring to mind some portion of the map, and "read off" the names of the states moving gradually from old to new portions of the map. No matter what mnemonic is used, the essentials are the same. In each case, the
to-be-remembered elements must be connected during storage to the elements in the mnemonic by some rule. Later, during retrieval this rule is applied.

To demonstrate the generality of this view of memory, consider what is involved in memorizing (learning) a new idea in mathematics—for example, "A function continuous over a closed interval is uniformly continuous over that interval." In this case, we may assume that the subject can interpret the statement (cf. Chapter 7, Section 6) so that the implied goal during storage is to find a rule which can be used to generate the indicated meaning from elements already in M—for example, from the meanings of individual signs, like "function." Rules which satisfy this goal may range from general logical inference rules, to manipulations on diagrammatic representations (of the meanings), to simple associations between the elements.

Suppose that sometime after the subject has stored the idea, he is asked something like "What can be said about functions which are continuous over closed intervals?" Here, the goal might be denoted: Find R such that R satisfies \( (S_0, G) \), where \( S_0 \) refers to "functions continuous over closed intervals," and R is a statement equivalent (in meaning) to "It is uniformly continuous." (More generally, to be acceptable as a response in this situation, R must be a property of functions which are continuous over closed intervals; uniform continuity is one such property.)

In order to generate an appropriate response in this case, the subject must have available some way of generating "uniform continuity," or a statement of some other allowable property, from "continuous functions over closed intervals." As with storage, the possibilities range from direct association to logical argument.

The instructions given in most memory experiments allow the subject a good deal of leeway in deciding how to store the to-be-recalled information. When we say "Learn statement A," for example, we frequently do not distinguish between the statement itself and its meaning. Tests used at recall, however, may either explicitly or implicitly make the distinction. Asking whether or not the subject has seen a particular statement before, for example, refers to the original statement, and not to its meaning.

---

5In Chapter 7, Section 6, recall, the goal was to generate the meanings of given statements. To do this, the subject must have available a suitable interpretation rule, or combination of rules (cf. Chapter 7, Section 6). A typical interpretation rule may require that the subject know the meanings of "continuous function," "closed interval," and "uniformly continuous function." He may also need to know the necessary grammar. What he does not know initially (before interpretation) is how these meanings are (linguistically) tied together. Interpreting the given statement amounts to integrating this knowledge by applying (or deriving and applying) an appropriate interpretation rule.
(cf. Bransford & Franks, 1971). Experiments run under such conditions, then, have a certain ambiguity.

Such ambiguity can be avoided in memory experiments by making the goal conditions in effect during storage compatible with those in effect during retrieval. For example, if a subject is asked to learn the meaning of a new statement during storage, then it would be inappropriate at recall to ask for an exact replica of the statement. (In that case, success would be expected only to the extent that the subject may have also memorized the statement verbatim during storage.)

Even more important, the memory theory provides a basis for explaining such ambiguity. Where the subject may use two or more rules to store or retrieve information, the motivation mechanism of Chapter 8 provides a basis for separation. Specifically, selection rules may operate at storage and/or at recall. Thus, certain subjects may tend to store "meanings" whenever possible, rather than to memorize statements verbatim, whereas other subjects may do the opposite. Parenthetically, we note that the recent results of Bransford & Franks (1971) at Minnesota suggest that the former is more common with college subjects.

Rule selection is involved in any memory situation where the criteria used by the experimenter to evaluate responses at recall are more specific than those (criteria) needed to define the task. Suppose, say, that in our mathematics example above, the subject is simply asked to recall the "uniform continuity theorem." Assume also that the subject "knows" both the statement "If a function is continuous over a closed interval, then it is uniformly continuous" itself, and also its meaning. That is, we assume that the subject is able to retrieve the theorem, either by parroting back the statement or by paraphrasing its meaning. The theorist's task under such circumstances is to explain and/or predict which kind of statement the subject will give. That is, he must be able to tell which retrieval rule the subject will select and why. In order to accomplish this, he must know or have some way of determining the selection rules governing the subject's behavior; this in turn specifies the conditions at recall under which one or another of the possible retrieval rules will be applied. It is worth noting that when we speak of rule selection we are, in fact, talking about something very much akin to searching memory for storage locations. (The locations correspond roughly to rules.)

More specific refined predictions concerning memory, then, necessarily depend on evoking the mechanisms of motivation discussed in Chapter 8. The predictions are somewhat more complex than those discussed earlier, however, since there frequently is no direct behavioral information
Further Analyses. Some additional examples may be of interest.

(1) Suppose that the subject's goal is to memorize the lines of a poem (rather than to determine their meanings). That is, the subject is required simply to commit the corresponding sequences of phonemes to memory much as young children are often taught songs, prayers, or the pledge of allegiance. Presumably, this is accomplished by determining structural descriptions of the given strings of phonemes (cf. Chomsky, 1963, 313-319). Determining these structural descriptions, according to our analysis, involves tying together via linguistic rules phonemes which are already in memory. (The structural descriptions, of course, could simultaneously be tied to some predetermined signal to recall--e.g., the name of the poem.) At recall, then, one could easily envision "priming" the subject with certain of the phonemes as a signal to recall the rest. (This, in fact, is what many teachers do in "getting a child started."

In order to retrieve a memorized poem, the subject must have available, or be able to derive, a rule for generating the desired sequence of phonemes. Fortunately, nothing basically new seems to be required here; there is a direct complementary relationship between constructing a structural description of a given statement and generating the statement. The same processes are involved, although in reverse directions.

The only contemporary memory research with which I am familiar that satisfies (approximately) the test conditions required by the theory in this section is the recently rediscovered memory technique of embedding to-be-remembered information in a prelearned sequence of images.6 Suppose, for example, that a person is to remember a list, say, of the fifty states by forming a "striking image" of each in an object (e.g., lamp, chair, etc.) in some previously well-learned sequence. In this case, the subject is typically given all of the time he needs to store each item. At recall, the person simply goes through the stimulus objects in the prelearned sequence, in turn, and "reads off" the names of the states.

In view of our discussion in Chapter 7 of the way symbols and icons denote, it also should be emphasized that our analysis is compatible with both verbal (symbolic) memory and imagery. It is just as easy to identify (storing) procedures for constructing rules which combine visual images as

6The ancient Greeks, as well as writers of popular books on "Improving Your Memory," seem to have been aware of this technique. Only recently, however, has the technique been systematically studied by experimental psychologists. The moral of the story, I suppose, is that at any given point in time there will always be things that work but for which we can find no scientific basis.
for constructing rules which combine symbols, or which combine symbols with images for that matter. Presumably, auditory (or other sensory) images, which correspond to melodies and rhythms of various sorts, can be dealt with in the same way. Only the details, and relative ease of analysis, may be expected to vary.

Interestingly enough, the same form of analysis seems applicable to certain kinds of verbal memory experiments as well, although any attempt to deal with specifics is beyond the scope of this discussion. Consider, for example, the task of remembering lists of nouns (e.g., mouse, fly, atom), some of which can be described by a common adjective (e.g., small). In this case, the to-be-remembered nouns must be tied during storage to some given signal to recall. To be successful, the subject must devise a rule during storage which generates all and only the to-be-remembered nouns. Unfortunately, the rules constructed during storage do not always adequately distinguish between to-be-remembered units, and others. For example, a rule involving concept categories (e.g., adjectives) may generate exemplars (e.g., nouns) of the concept other than those to be recalled. The resulting intrusions presumably interact with other constraints on the task (e.g., the number of exemplars to be recalled) to reduce the level of recall. For a detailed description of experiments that can be analyzed in this way see Scandura (1967c) and Scandura & Roughead (1967).

Conclusions, Implications, and Limitations. This completes our discussion of the memory theory with unbounded processing capacity as such. It would be nice at this point if there were direct empirical evidence to cite which supports the proposed theory, but unfortunately none yet exists.

There is, nonetheless, some intrinsic support. For one thing, the mechanisms of the memory theory are a natural extension of those on which the idealized theory is based. What goes under the rubric of memory can be handled by extending the basic mechanisms of the idealized theory only slightly to allow for the generation of domain elements. Moreover, the extension provides a framework within which to view the relative roles and interrelationships among such disparate phenomena as simple performance, problem solving, learning, and motivation, not to mention long-term memory. In an important sense, the mechanisms of the memory theory are as simple as those required in most existing theories designed to deal with specific phenomena (e.g., memory), and yet, the theory has much greater generality. Indeed, in the formalization of the theory presented in Chapters 5 and 9, it appears that the same general mechanism may also provide a basis for dealing with perceptual growth, and indeed, human development generally.
Another line of support for this memory theory is that the basic mechanisms seem to provide a basis for explaining a number of common phenomena related to memory. In accord with known facts, for example, it follows directly that degree of recall is generally dependent on the extent to which the test conditions reinstate the stimulus conditions during storage. In addition, the distinction between goals and inputs (stimuli) provides a basis for making finer experimental distinctions than has for the most part been the case to date.

Arousal phenomena provide another area of application. According to the theoretical mechanisms proposed, it is immediately obvious why a rule that has been used in the immediate past is more likely to be used than some alternative (rule), even where, as in Einstellung experiments, the alternative would otherwise be preferable. Rules in A are applied before rules which must be derived (retrieved) from rules in A.

Similar comments can be made concerning retroactive inhibition, and even reminiscence. This can be seen by simply observing that learning may take place between the time information is stored and retrieved and that, in general, this new learning may either interfere with or facilitate retention. (Note that this view lumps together traditional views on interference theory and trace theory (cf. Osgood, 1953, 587-599).)

As an example of how additional learning may facilitate retention, consider the formula \( \csc^2 \theta = 1 + \cot^2 \theta \) from trigonometry. Suppose that in the interim period between learning this formula and recall the subject is trained on the Pythagorean Theorem (i.e., \( A^2 + B^2 = C^2 \), where \( A, B \) and \( C \) correspond to sides of a right triangle), and shown how the trigonometric identity \( \cos^2 \theta + \sin^2 \theta = 1 \) may be obtained from the Pythagorean Theorem by dividing each term of \( A^2 + B^2 = C^2 \) by \( C^2 \). It is also assumed that the subject is able to compute and know the definitions of the various trigonometric functions (e.g., \( \cos \theta = B/C \)). Then, one might reasonably expect this interim learning to have a facilitating effect on retention of the original formula \( \csc^2 \theta = 1 + \cot^2 \theta \) even though the intermittent training did not directly involve that formula. Presumably, what would happen in this case, is that some subjects would be unable to retrieve the formula directly but be able to derive the formula from the

---

7In most experiments, activities between presentation and recall have had a detrimental effect on retention. Research in the area has been aimed primarily at determining the relative amounts of inhibition associated with different kinds of activity. In a few cases, however, retention has been shown to increase over time. These phenomena have been referred to as "retroactive inhibition" and "reminiscence," respectively, and the preferred explanations have involved interference and consolidation.
In the case of retroactive inhibition, we can refer again to one of my studies (Scandura, 1967c). In this study, the experimental subjects were encouraged to classify lists of nouns according to properties which the objects denoted by these nouns shared in common (e.g., "mouse, atom, fly" went with "small")—that is, to construct rules during storage that involve such concept defining properties. It was found that the retroactive presentation of additional nouns, which were associated with the same concept categories as the to-be-remembered nouns, interfered with the retention of the original nouns. Intrusion data further suggested that the intermediate nouns were stored in the same concept categories (locations) as were the original nouns, thereby apparently confusing some of the subjects at recall. Apparently some of the subjects were unable to distinguish between those nouns placed in the categories originally and those placed there shortly thereafter.

Although this experiment provides general support for the analysis, it is important to emphasize that the theory itself is essentially deterministic while the experimental results were based on averages. Any critical test of the proposed theory will necessarily have to be based on more highly structured tasks, tasks where storage and retrieval rules may be isolated for study. The lack of explicit structure in most traditional experimental tasks allows for such great individual differences as to essentially preclude attempts at definitive analysis.

More generally, a basic problem in checking the present theory against most contemporary memory experiments is that little is known about the kinds of storage and retrieval rules subjects typically use, let alone the ones they will use in particular instances. This is partly due to the intrinsic difficulty of the problem. But, the blame cannot all be put there. Although purportedly simpler than complex (structured) learning, for example, the experimental tasks typically used in traditional studies are generally much more difficult to deal with. The tasks are artificial and highly unstructured and very little can be said beforehand about how a subject might attempt to store the to-be-remembered material or to retrieve it once it has been stored. There are apt to be as many different procedures involved as there are individuals and items to be remembered.

Given a list of nonsense trigrams (e.g., XPT), for example, it is not at all clear how a particular subject might go about storing them. One subject might attempt to "key," say, on first letters and another on form. The situation with structured materials is generally simpler because people are more inclined to store the material in much the same way. It is not
likely, for example, that during storage a person would relate "4 + 5" to "2", "1+", and "5". He would be far more likely to use "4", "4+", and "5". The situation frequently is only slightly better in studies concerned with more conceptual material. In another of my earlier studies (Scandura & Roughead, 1967), for example, at least some of the subjects seemed to prefer storing the nouns on the basis of the cadence or rhythm imposed on them during their initial presentation (rather than the concept categories presented).

Needless to say, it would be far more difficult in these types of situations than in highly structured ones to identify the kinds of storing and retrieval procedures which subjects might use. If we want to make memory processes the object of serious scientific study, such studies in my opinion are not the place to begin. At least initially, the tasks used should be such that we can be pretty sure what storage and retrieval rules are likely to be used.

Equally important, although they have surprising power for their simplicity, the mechanisms described above are not adequate in themselves to give a complete account of either human behavior in general or memory in particular. The theory by definition avoids questions relating to time (e.g., presentation time) and processing capacity, and thereby has a reduced domain of applicability. A general theory of memory, for example, would have to account for the fact that subjects are not always able to store all of the information they are presented with even where they have adequate storage procedures available. In particular, too much information may be presented at too rapid a rate so that only some of it can be processed and, thus, stored. To provide such an account, the above memory theory would have to be extended to explicitly deal with the time required at each stage of a computation.

Experiments involving heavy processing loads, whether or not memory is at issue, are also beyond the scope of the theory. Thus, for example, according to the theory a person who can add can add all numbers, large ones as well as small ones. The fact that most human beings can perform mental computations with some numbers but not others cannot be handled. The theory simply does not apply where the information processor (subject) is limited in the amount of information he can process at one time. Parallel comments would apply to short-term memory experiments where the subject must retain in A (perhaps through rehearsal) a large number of elements. Interpreting (i.e., storing) sentences like "The rat (the cat (the dog chased) ate) sat," where simple sentences are successively embedded in one another, involves similar limitations. In practice, there
will be only so many such embeddings that a human subject can handle.\footnote{Presentation time in this view is \textit{not} a primary variable. Its well-known effects on performance can be explained by observing that information may arrive at too fast a rate for adequate processing (i.e., storage). In particular, the subject's information processor may easily be overloaded with new information while still attempting to process (store) the old.}

Furthermore, as we shall see in Section 2.2, allowing the subject to use paper and pencil (or other memory aid) effectively results in a different rule being used. And although this may effectively improve the subject's performance, it does not change his fixed capacity to process information.

### 2.2 Memory Theory with Fixed Processing Capacity

The mechanisms of the memory theory with unlimited processing capacity can serve only to make more rules in M active. Mechanisms are also needed to explain how information is deactivated. In the absence of relevant data, it is not clear how such mechanisms might operate. There are, however, two basic ways in which deactivation might enter the theory: (1) by modifying the basic mechanism so as to allow for deactivation of goals as well as activation, and (2) by modifying the rule notion itself so that elements may be "erased" as well as generated. The basic constraint in either case, according to the present view, is the fixed finite capacity of $A$.

Insofar as the basic learning mechanism is concerned, a reasonable assumption to make is that goals are deactivated when they become no longer useful in a derivation. One possibility might be that goals (at various levels) are retained in $A$ only where they are needed to determine some future goal in the derivation. Thus, for example, any given $G$ will remain in $A$ throughout the course of a derivation, because it is always the last (as well as the first) goal in control. Higher order goals (e.g., $G^2$, $G^{2\text{Dom}}$), however, are discharged from $A$ as control reverts to lower levels. Retaining them in $A$ after this would serve no useful purpose. (In order to avoid possible confusion on this point, my last statement is not to be construed as support for the proposed mechanism, but rather as a rationalization. In this regard, it is my firm belief that a viable theory should have an overall unity and simplicity, with a minimum of \textit{ad hoc} assumptions. Discharging goals from $A$ as indicated, would tend to satisfy this general criterion.)

Specifying how goals become active and are deactivated, of course, tells only part of the story. It still leaves open what happens, for example, when a rule becomes overloaded in the course of a computation. In this case, one might make some general assumption, such as: The element in $A$ that was processed first (e.g., seven elements ago) is always dropped...
first. (In fact, the primacy effect would suggest just the reverse.) Such assumptions, however, would be subject to many of the same limitations as the experience-free hypotheses of Chapter 8 and are not seriously considered here. In particular, according to the present view, the basis for deactivation (erasure) will vary from time to time, from rule to rule, and from subject to subject.

We prefer instead to add more structure to our rules, so that they not only activate (generate) new elements during the course of a computation, but also deactivate (erase) others. In this case, rules must not only specify what is to be done at each stage of a computation, but also where each generated element is to be located. Think of the problem this way: The subject has, for example, seven labeled spaces in his processor. One space, for example the first, might contain the goal (the stimulus is in the environment); the other spaces are filled with rules and elements (which do not operate on rules). Control automatically shifts to the higher order goal; whatever is in location 2 is erased from A. Then A is automatically "searched" for that (unique) member of A (if any) that satisfies the higher order goal, and control shifts if necessary as before. In applying a rule, we specify both the elements generated at each stage and the location where these elements are to be stored. The elements already in these locations are "erased" (i.e., removed from A).

Perhaps this is all that can be said. The locations vary from time to time and rule to rule, with few if any constraints. If this proves to be the case through empirical investigation, then the number of possibilities will probably be so large as to necessitate a stochastic solution (theory) at this level.

For this reason, it obviously would be premature at this point to settle on any one set of constraints. Nonetheless, it is interesting to speculate about some of the more obvious possibilities. A rather mechanistic solution would be to propose that the spaces be ordered (as well as labeled) and that newly generated elements be stored, say, in a cyclic manner. That is, each new element that is added, replaces the element in the "next" space.

Although some such procedure might be convenient if one were to build a computer analogue, I doubt that this type of solution would hold up empirically. As we shall see in the third (empirical) section, there is strong reason to believe that subjects can routinely eliminate elements according to need. That is, subjects tend to retain information necessary at some later point in a computation, but to discard information that is no longer useful. Clearly there is no direct relationship between cyclic
storage and the latter.

In effect, we are proposing that the rationale imposed above for discharging goals from $A$ might be extended to rules and elements in $A$ as well. The basic idea is this. At each stage of a computation (i.e., application of a rule to a given stimulus), a specific number of elements must be retained in $A$ in order to determine future outputs and operations. At the conclusion of any given stage, then, it may be possible to erase certain of the elements required at a preceding stage. In adding 45 plus 71, for example, it is necessary to remember the two units digits 5 and 1 before summing them; but afterward only the partial sum 6 must be retained.

The number of elements in $A$ at any given stage of a computation constitutes the memory load at that stage. The memory load of a computation is the maximum of the memory loads associated with the various stages (of the computation). Details as to how memory loads are determined for individual rules are given in Section 2.3.

Parenthetically, it is important not to confuse memory loads at particular stages of a computation with states (i.e., equivalence classes of past histories of stimulation) of an automaton. Memory load is the number of elements needed to define a state, plus whatever other elements are needed to determine future outputs. Although the number of elements needed to define a state remains constant throughout a computation, the number of additional elements may change where looping is required. The memory load, therefore, may change each time the computation goes through a loop.

The memory theory with fixed capacity is much like a finite automaton in which the elements involved may themselves be finite automata. In this case, the original automaton must have some mechanism which tells how its elements are used. Although such an arrangement may prove to have essentially the same intrinsic computing power as an ordinary finite automaton, it is not mathematical equivalence that is of primary concern. The problem is to reflect behavioral reality.

To summarize, we are proposing that elements in a computation are not only generated, and thereby added to $A$, according to predetermined laws, but also erased in a specified manner. The same general principle applies to the shifting of control among related goals. Indeed, in principle, the same priorities might be extended to encompass entire learning episodes (i.e., from initial goal situations to their resolutions). The elements in a learning episode, whether they be simple elements, rules, or goals,

---

9Indeed, we talked above only about discharging goals, and said nothing about where new goals were to be stored.
are retained in $A$ if and only if they are needed to determine either a future output or an operation that must be performed some time in the future. (At this point it seems safest to leave open how far into the future the subject may anticipate.)

Some feeling of how this might go can be obtained by considering an example of Type 1 in Section 2.1. Specifically, let $A = \{S^o, r_n, r_m, o, G\}$. Control shifts to $G^2$ so that now $A = \{S^o, r_n, r_m, o, G, G^2\}$. (If $A$ becomes overloaded at this point, something crucial must be erased, but our theory does not specify what.) Here, $o$ is applied to $(r_n, r_m)$ generating $r_m \cdot r_n$. This time, however, instead of just adding $r_m \cdot r_n$ to $A$, (at least one of) $r_n$ and $r_m$ may be erased, as they are no longer needed. Similarly, once control reverts to $G$, $G^2$ is erased, leaving "more space" for the application of $r_m \cdot r_n$.

Finally, we just mention in passing the question of processing time. Rules obviously take time to apply, and any complete theory will want to deal with this fact. This is not a new problem for psychology, of course. A number of experimental techniques have been developed to estimate processing time. See Sternberg (1969) for a recent summary and extension of traditional methods. Although it is beyond the scope of this book to consider this problem (except obliquely in Section 2.3), any ultimate solution must clearly transcend methodology. It is my belief that processing time may ultimately be traced to certain physiologically based behavior constants of individual subjects.

To say more about the general theory at this point would be unwarranted. Indeed, the hard-headed empiricist will undoubtedly feel that too much has been said already. The next section deals more intensively with one facet of the problem for which empirical evidence has been obtained. In the process, the preceding analysis is extended and a new experimental paradigm is proposed and tested.

2.3 A New Experimental Paradigm

Testing the memory theory with unlimited processing capacity has much in common with the memory-free paradigm. Extension to the case with fixed processing capacity, however, is not so simple. Empirical tests of basic mechanisms in the latter case will almost certainly require the development of new and more rigorous experimental paradigms.

---

The research reported in this section was conducted in collaboration with Donald Voorhies.
One step in this direction is taken below. Specifically, we develop and test an analytic technique for calculating the memory load attached to specific rules in the course of computation. Stochastic models are also proposed in order to account for deviations between analytically based predictions and the data. Then, we suggest briefly ways in which this paradigm may be extended. We begin with a brief review of the related research.

**Related Research.** The distinction between primary and secondary (permanent) memory has been with us since William James. It was not, however, until Miller's (1956) classic work that the notion of a fixed processing capacity came into prominence. Synthesizing research involving absolute judgments, span of attention, and span of immediate memory, Miller showed that humans have a definite limit in the amount of information they are able to process at one time, and that generally speaking the limit is 7 ± 2 elements.

Drachman and Zaks (1967) further demonstrated that there is a rather sudden decrement in performance—a "memory cliff"—at capacity. They administered the same task twice, using the data from the first administration to account and adjust for individual variation on the second. After first determining individual memory spans, they administered additional memoranda (strings of digits) of various lengths. The dependent measure on the second test was a function of the digits recalled by subjects on strings of length \( \text{span minus one}, \text{span}, \text{span plus one}, \text{etc.} \) Pooling the data across subjects, and adjusting for individual spans in this way, there was a fairly sharp break in performance beyond the estimated memory span.

This limit appears to be independent of whether or not the elements being processed are familiar. Crowder (1967) and Conrad (1960), for example, found that requiring subjects to prefix to-be-remembered lists at recall with an extra, redundant digit (e.g., "1") causes a decrement in the percentage of correct responses. Savin (1968) further showed that prefixing a given digit to, say, an eight digit list causes the list to act at recall as a nine digit list. The extra digit, even though highly familiar to the subjects, apparently occupies as much processing space as any other.

Processing capacity, however, depends only indirectly on the number of elements. As Miller (1956) has shown, the essential factor is the number of effective units, which he called "chunks." In a similar vein, Bower (1969) has shown that the critical feature in recall is the number of functional units in memory, rather than the sheer number of units. He had subjects recall words from lists which included single words (one
functional unit), three-word cliches (one functional unit), or three-word triples (three functional units). In all cases, although the number of words recalled varied, the number of functional units recalled was the same.

More important in view of the approach adopted below, there are indications in the literature that the number of chunks depends on the procedures subjects use as well as on the tasks themselves. Posner and Rossman (1965), for example, gave subjects number strings, and required them to transform the strings in various ways (e.g., add, give the highest number, give the lowest number, etc.). They found that both the number of transformations and their size had a significant effect on the amount of untransformed material that was lost from store. The more processing required of the subject, the less peripheral material he can continue to rehearse. This suggests that on the average the more complex a procedure, the more processing space is required to apply it. (In this regard, it should be noted that Posner and Rossman used an information theoretic measure of transformation difficulty (i.e., size of reduction in "bits"), which on some tasks was not compatible with obtained difficulty. We comment on this again below.)

This dependency, clearly, is not a simple one; procedures can serve to decrease the number of chunks that must be retained, leave it unchanged, or possibly even increase the number. The results of Miller (1956), for example, suggest that a subject can increase the absolute number of elements (e.g., strings of binary digits) he can process by recoding them into a smaller number of chunks (e.g., strings of decimal digits) in a way which makes it possible to regenerate the original elements on demand. Dalrymple-Alford (1967) overcame the highly variable and often idiosyncratic effects of rehearsal by building it (rehearsal) directly into his experimental procedure. Rather than attempt to prevent it, he tried to insure that rehearsal was of a known type and magnitude, and was uniform across all subjects. He had his subjects rehearse all previous digits aloud before the experimenter added a new digit. For example, suppose the first four digits were 2, 7, 3, and 9. Then, the subject had to say "two, seven, three, nine" before the experimenter presented the next digit. Following Brown (1958), Dalrymple-Alford reasoned that any increase in retention as a result of rehearsal is due to recoding and not rehearsal per se (i.e., the "strengthening of memory traces"). With this in mind, he also determined each subject's memory span in the conventional (Woodworth & Schlosberg, 1954) manner. Surprisingly, the proportions of
errorless repetitions were almost identical under both conditions indicating that no learning (recoding) occurred during rehearsal. Dalrymple-Alford further noted the absence of traditional serial position effects. First errors in repetitions (breakdowns) tended to occur equally often at all positions of the number sequences.

The limited capacity hypothesis (cf. Broadbent, 1958) also has considerable explanatory power. It has been used as a basis for explaining and/or predicting the effects of a number of variables in short term memory experiments.

Katz (1968), for example, found that primacy and recency effects on the serial position curve for short-term free recall are due to the subject's selective use of attention and storage and retrieval strategies operating under the constraint of a fixed short-term processing capacity. He had his subjects repeat random lists of two-digit numbers, and instructed one group to give the last pair before recalling the others. The other group was allowed to give the digits in any order they wished. Although he found no overall difference between the two groups in correct recall, the former group showed a relatively strong recency effect and the latter, a relatively strong primacy effect.

The results of several other studies suggest that processing capacity underlies the opposing effects of certain pairs of factors. Murdock (1965), for example, reasoned that if memory capacity is constant, then it should be possible to "trade off" the number of times a stimulus is presented with the exposure time on each trial. He constructed a list of six pairs of common English words, and presented them for study for a period of 24 seconds. Three conditions were used: (a) In the first condition, each pair was presented once, for a period of 4 seconds; (b) in the second, each pair was presented twice for a period of 2 seconds (on each trial); (c) in the third, each pair was presented four times for a period of 1 second (on each trial). Murdock found that there was essentially no difference between conditions a and c, with b only slightly better than both. In a study involving incidental learning, Somers (1967) obtained basically the same result. Sitterly (1968) used various combinations of digit and interdigit duration and also found that retention depended on total presentation time.11

11Other studies have been of a primarily empirical nature. Thus, for example, Williams and Fish (1965) showed that increasing the length of the individual items to be recalled, and increasing the number of symbols from which the items are constructed, both decrease the percentage of correct recall (of the items). Corballis (1966) varied the speed at which strings of nine digits were presented as well as the duration of time each digit was actually visible. He found that when stimulus durations (continued)
Use of the limited capacity hypothesis as a basis for explanation with more complex tasks has been limited largely to correlational studies. Whimbey, Fishchhof, and Silikowitz (1969), for example, measured performance on a digit span task, a mental addition task, a vocabulary task, and a test of general intelligence. They found a generally high correlation between the digit span task and the mental addition task, suggesting that both are influenced by a limited processing capacity. Vocabulary and general intelligence both showed little relation. Whimbey and Leiblum (1967) found similar high correlations among simple repetition, repetition preceded by "0," and repetition with interspersed verbal activity. Later, Whimbey and Ryan (1970) found correlations between the digit span task and mental syllogistic reasoning problems.

By way of summary, three general conclusions may be drawn from the available literature. First, there is a definite limit on the number of effective units (chunks) that subjects may process at any one time; this limit is 7 ± 2 chunks. Second, the memory load imposed by a task depends in a nonsimple way on the procedure used by the subject as well as on the task itself. Third, the limited capacity hypothesis provides a useful basis for explanation and, hence, might well be expected to play a central role in any viable theory of memory.

Although more might be gleaned from the available literature, there are two important limitations inherent in it. For one thing, there is no way to specify precisely how a subject is to process given information, except with the simplest tasks. Savin (1968), for example, required his subjects to prefix a given digit at recall but did not specify how this digit was to be processed in relation to the others. Other investigators have instructed their subjects not to organize the given digits but made no attempt to tell them what they should do. Under these conditions it is uncertain, for example, when information is in memory and when it is in the information processor (Waugh & Norman, 1965). Even on simple tests of memory span, recoding and rehearsal processes tend to be highly dependent on individual preferences and susceptible, at best in only partly known ways, to even slight changes in experimental conditions. Indeed, in only one study reviewed (Dalrymple-Alford, 1967) was a serious attempt made to insure that the subjects used a particular procedure (rehearsal).

II (cont'd.) were long, the number correct was higher the slower the presentation speed, but when stimulus durations were short, there was a tendency for this trend to be reversed. Although generally compatible with the limited capacity hypothesis, the results were relatively complex and no theoretical explanation was offered.
For another thing, even if procedures were to be specified, there is no analytic method available for determining the processing load they (the procedures) impose on the subject. Under these conditions it has been impossible to compare different tasks (procedures) with regard to memory load, short of direct empirical test. As promising as the technique originally appeared, for example, measures of information reduction (e.g., Posner, 1964) seem to work well with only certain tasks. Suppes (1967) proposed an analytic method for calculating memory loads that was used with some success in predicting latencies on simple addition problems, but the method was too crude for present purposes.

Until a way is found to overcome these problems it will be difficult if not impossible to adequately test the proposed theory. Although it seems safe to assume that subjects differ in their ability to process information, for example, it is not clear whether these differences are due to innate physiologically-based capacities or to the characteristic use of particular kinds of procedures. There is no way of distinguishing between these possibilities in available studies. A conclusive answer to this question could have important implications for the form the proposed theory should take. In particular, if the ability to process information is independent of underlying physiology, then in the theory it would be well to assume that the capacity of the processor $A$ is the same for all subjects. If it is not, then the capacity of $A$ might better be allowed to vary over individuals. Furthermore, there has been no way short of direct empirical test to determine expected performance levels on specific tasks (procedures). Explanation and prediction via a competence-based theory of the sort proposed will require more exacting analytic methods.

Objectives. The ultimate aim of this research is to develop an experimental paradigm suitable for obtaining more definitive information concerning the limited capacity hypothesis, and its role in memory. More immediately, we wanted to develop and test (1) an effective method for instructing subjects in how to process information according to specific procedures, and (2) an analytic method for determining the memory loads imposed on subjects by given procedures as applied in particular task situations.

Suppes' (1967) regression model involved three structural variables, the magnitude of the sum (MAGSUM), the magnitude of the smallest addend (MAGSMALL), and the number of steps necessary to complete a problem (NSTEPS). In determining NSTEPS, specific account is taken of how many quantities must be kept in memory in the course of solving a problem. The predictions made by the model correlated .86 with the actual data of 24 fourth-grade students.
In the process we hoped to add to our store of information concerning the limited capacity hypothesis. Specifically, we wanted to find out whether this capacity is based directly on the underlying physiology of the organism or the characteristic use of particular kinds of procedures. In the latter case, presumably, "capacity" would be subject to training. We also wanted to get some feeling for the suitability of various tasks for such research. To date, almost all of the experimental (as opposed to correlational) research has been done with some form of list learning. Such tasks are so restrictive that they may not by their very nature involve certain parameters of general concern. This possibility would be minimized by using a task, say, like addition.

Incidentally, we also wanted to obtain further information concerning rehearsal. Does it improve memory directly (by strengthening associations) or only indirectly by increasing the opportunity for chunking?

In accordance with the basic objectives, the study had both an analytical and an experimental component.

Phase I. During Phase I, which began during November, 1969 and ended roughly during the summer of 1970, a number of informal pilot experiments were conducted in order to gain some feeling for the phenomenon. Noncarry addition was chosen as an initial task and several methods of presentation were tried out. It quickly became apparent that the usual tabular form (e.g., \( \begin{array}{c}
452 \\
+ 346
\end{array} \)) made it difficult to get subjects to process the information digit by digit as was desired. To help overcome this problem, a linear form of visual presentation was devised in which the digits were typed vertically on index cards (e.g., 2 6 5 4 4 3). The subjects were asked to read the digits aloud, in turn, and to add where appropriate. (The individual column partial additions are indicated by slash marks.)

A similar mode of presentation was developed for: (a) addition with carrying (with carrying involved after each partial addition), (b) multiplication with a single digit multiplier, and (c) three closely related tasks involving digit lists—repeating the digits in lists in the order given (L), saying "one" before repeating the digits (1 L), and saying "one" after repeating the digits (L 1).

The subjects were instructed to avoid coding, mnemonics, or using rhythmic patterns in reading the numbers, as this tended to encourage idiosyncratic "chunking," thereby making it difficult to test any analytic technique for calculating memory load. On the positive side, about all we could tell the subjects was to maintain a uniform rhythm in processing
the numbers, to give equal emphasis to each digit, and to treat them as separate entities never consciously linking them in any way.

During the course of these pilot experiments, we tried a number of different methods (analytic techniques) for calculating memory load. At first we tried simply counting the number of digits that had to be retained at each stage of the computation. In noncarry addition, for example, after the first two digits (i.e., 2,6) have been presented, the load is two. After these two digits are added, the load is one, the partial sum 8, because the two addends are no longer relevant and may be discarded. The third and fourth digits (i.e., 5,4) raise the memory load to three, until addition discharges these digits and replaces them with the second partial sum (i.e., 9), giving a load of two.

The maximum of the loads determined during the course of a computation, involving a particular problem and rule, constituted the memory load for that task (i.e., rule and problem). In the three-digit non-carry addition problem above, the memory load was four as regards the indicated addition rule. This maximum load is obtained after the first two partial sums have been determined and after the last two digits have been encoded. After the third (partial) addition, notice that the load goes down to three.

Although it seemed to provide a reasonable first approximation to the truth, this simple analytic technique was not adequate. Memory loads determined by the technique for adding n-digit numbers are the same for noncarry and for carry addition; but our pilot data showed that subjects could almost invariably add larger numbers (e.g., having one more digit) when carrying was not involved than when it was. Moreover, the data suggested that our college subjects had processing capacities of the order of four or five, rather than the seven + two chunks found by Miller (1956).

For this reason a more complicated technique was developed in which operators (e.g., the act of encoding or of adding two digits to determine a partial sum), as well as states (digits), were counted whenever they were needed to determine a subsequent state or operator. In noncarry addition, for example, encoding two digits is invariably followed by an addition operation. In this technique the number of distinct categories (sets) of digits involved was also counted. Since partial sums, for example, play a different role in noncarry addition than do addends, they were viewed as separated in memory. Specifically, the load at each stage of a computation was determined by adding the number of: (1) states
needed to determine subsequent states or operators, (2) operators needed to determine subsequent states or operators, and (3) different categories of digits.

Although this method brought processing capacities more in line with our expectations, it also, unfortunately, appeared to have serious faults. According to the technique, a subject who can do noncarry addition with two three-digit numerals should be able to repeat back a digit list of length eight. Both tasks impose a memory load of eight according to the analytic technique. The data, however, did not support this analysis. The analytic technique was similarly in error in comparing multiplication and carry addition.

Equally important, the data were not nearly as clear cut as we had hoped. In spite of the special precautions taken, the sharp drop (memory cliff) in performance expected in moving from a task just within a subject's capacity to one just beyond, was realized in only between 30% - 40% of the cases. Memory cliffs existed in the other cases, to be sure; but they tended to be more gradual, as with the Drachman and Zaks data, than we had hoped.

One of the major problems in this regard was that our instructions consisted largely of telling the subjects what not to do rather than exactly how they were supposed to process the information. Furthermore, our experimental technique was rather inefficient in that the size of each problem (e.g., the number of digits in the numbers being added) was fixed in advance, thereby limiting the amount of information that could be obtained on each trial.

**Phase II.** At about this time the author hit on the idea of using a modified form of directed graph, in which nodes and arrows are labeled, as a basis both for calculating memory load and for instructing the subject in how to process the information. The experimental technique was also modified in order to achieve more efficient information gathering and better monitoring of the subjects' performance during processing.

**Analytic Technique.** The analytic technique evolved from my contention that there is a fixed level beyond which any given rule (labeled directed graph) cannot be refined (cf. Chapter 5). More exactly, the stimuli and responses in the extension of a rule are constructed from atomic elements. These atomic elements impose a maximum level of detail which cannot be exceeded by any refinement of the rule.

Consider, for example, column addition without carrying. To make things definite, consider the stimulus $56 +_{23}$, and the usual addition algorithm in which the digits in the respective columns are added in turn,
beginning on the right. The first step in the algorithm would be to encode the top digit (6), and then, the bottom digit (3). Next, the two digits would be added, giving 9. Control would then move to the next column and the process would be repeated. After each column has been added, the final step would be to state the digits in the sum out loud. This algorithm can be represented by the labeled directed graph

The important thing to notice about this algorithm is that any further refinement would necessarily require redefinition of the digits themselves (e.g., in terms of their meanings).\(^\text{13}\)

To show that a maximal refinement exists in general is not difficult, but I shall only sketch the argument here. Note first that every rule has at most a finite number of operations and decision making capabilities. The gist of the argument is that although we allow operations with infinite domains for efficiency in representation, the very notion of effective procedure requires that each output must be generable by a finite number of finite operations (e.g., associations) on a finite number of finite input arrays. Effectively, this is equivalent to saying that any procedure can be reduced to a Turing machine. Thus, for example, although individual addition problems (stimulus arrays) may be of arbitrary size, the steps in the addition algorithm itself ultimately depend only on a finite number of different (and finite) substimuli, in this case those which correspond to the basic addition facts. The essential point is that the substimuli, which determine outputs, are necessarily composed of atomic elements, so that there will always be some maximum level of detail.

In the memory-free theory, the level at which we were willing to represent a rule depended on what could be assumed to be atomic; operations with infinite domains were commonplace. In dealing with memory and information processing, on the other hand, the level of refinement must be absolute. Assumptions are made concerning which input and output elements are atomic (i.e., directly perceived and indivisible), and each step is assumed to be a simple

\(^{13}\) We could, of course, add extraneous associations, say, between pairs of digits in given columns and their sums. This special type of refinement would be nonessential, however, because no further refinement of decision making capabilities in the algorithm is involved.
association—all intermediate steps must be represented explicitly. Representation even at this level of detail, however, does not tell the whole story. What is missing, essentially, is some way to represent memory load.

In information processing, substimuli of a stimulus display are encoded in turn, and subresponses are generated. During the course of a computation, new elements may be encoded and old ones erased. Specifically, each encoding operation serves to encode finite substimuli; this adds to the memory load. Each operation serves to generate new subelements (encoded substimuli) and/or to eliminate others. Each decoding operation serves to decode subelements as subresponses; this reduces memory load.

As a result of fumbling about with various methods of calculating memory load, many of which were largely equivalent, it appears that the load in a processor may be characterized most simply and effectively in terms of the number of distinct substimuli which must be dealt with in the course of a computation. It was hard to see this at first because:
(1) substimuli which are parts of other substimuli count just as much as if they were discrete and (2) new substimuli may be constructed during the course of a computation.14

The formal representation finally settled upon is basically a refinement of the labeled directed graph formulation of Chapter 5, in which each node is labeled with the substimuli that must be retained at each stage. For example, the above graph for noncarry addition can be represented

START

\[
\begin{align*}
\text{(read } a_i \text{)} & \quad a_i \\
\text{(read } b_i \text{)} & \quad b_i \\
\text{(add } a_i \text{ and } b_i \text{)} & \quad \left( s_1, s_2, \ldots, s_{i-1} \right) \\
\text{STOP} & \quad \left( s_j, s_{j+1}, \ldots, s_i \right) \\
\end{align*}
\]

\[\text{(read } \ast; \text{ Say } s_j, \quad j = 1, 2, \ldots, i; \quad \text{and shorten substimulus to } s_{j+1}, \ldots, s_i; \quad \text{set } j = j+1)\]

14 At first, these newly-constructed substimuli were called "chains."
For convenience, the nodes have been replaced with circles and the arrows represent the operations indicated in parentheses. The variables inside the circles refer to (encoded) substimuli as do the small ovals. The variables and the sign * outside the circles refer to new inputs. The use of indexing variables is essential because loads may change each time through a loop.\(^\text{15}\)

An entity counts as a substimulus if and only if it must be distinguished (referred to) during the course of a computation. The memory load at any given stage, then, is the sum of such substimuli; the memory load for a given computation is the maximum of the loads at the various stages. In adding three-digit numbers, for example, the load at decision making capability \(a_1\), \(s_1 - s_2, \ldots - s_{i-1}\) is five, two for the partial sums \(s_1\) and \(s_2\), one for the substimulus \(s_1 - s_2\), one for \(a_3\), and one for the stimulus \(b_3\). The maximum processing load of six is at decision making capability \(a_1, b_1, s_1 - s_2, \ldots - s_{i-1}\), two for the partial sums, one for the substimulus, two for the digits \(a_3\) and \(b_3\), and one for the substimulus \((a_3, b_3)\) which serves to elicit the partial sum \(s_3\). No overt stimulus is presented at this point; the operation is strictly internal. The load goes to five at decision making capability \(s_1 - s_2, \ldots - s_i\), three for the partial sums \(s_1, s_2,\) and \(s_3\), one for the substimulus, and one for * which indicates that the subarray is to be decoded. After \(s_1\) is discharged, the load goes to three, two for \(s_2\) and \(s_3\) and one for the subarray. The * is no longer needed. From there the load decreases monotonically to zero.

To summarize, the key idea is that the memory load at any given stage of a computation depends directly on the number of (encoded) substimuli, atomic or otherwise, that are required to determine a subsequent output in the computation.

**Method.** Six tasks were used in the experiment: (1) repeating digit lists (L), (2) repeating digit lists an extra time (XL), (3) saying "one" before repeating digit lists (1L), (4) noncarry addition (NCA), (5) carry addition (CA), and (6) mixed addition (MA) in which both carrying and noncarrying are involved.

In the three list tasks, the digits were presented orally one at a time. After each digit was presented, the subject was required to repeat

\(^{15}\) Although we refer to certain of the operations as "read," thereby implying some sort of encoding, we make no commitment in this regard. Although it is true that individual digits were read in turn to our subjects so that they had to encode (read) them, the operation can be viewed equally well as strictly internal, where the operation corresponds to selecting various parts of an encoded stimulus array.
every digit presented up to that point, saying "d_1 then d_2 then ... d_j then d_{i-1} then d_i."
where d_j is the jth digit and d_i is the last. After hearing the instruction "repeat"
in task XL, the subject was simply to repeat the last string of digits. In task IL, the subject had to say "one"
before repeating the last string. The lists used in tasks XL and IL were of predetermined length,
whereas those used in task L were not. In the later case, new digits were added until the subject made a mistake.

The addition tasks were presented in a similar manner. Each successive pair of digits corresponded to the digits
in one column of a column addition problem. After encoding each such pair, the subject was required to say
the sum of the two digits out loud. In the NCA task, the next pair of digits was presented immediately thereafter,
and the process was repeated until the subject made a mistake. In the CA task, the subject was required to verbally separate
the tens and units digits of each column sum (e.g., subjects would say, "17" and then "7, 1") before continuing,
and then to add one to a_j, the next digit presented (i.e., to carry). In the MA task, the tens digit of the sum was sometimes zero
in which case nothing was added to the next digit presented.

The processing procedure used with the NCA task can be represented by the node labelled directed graph shown below.
The other procedures used may be represented as follows.

**TASK CA**

```
START
a_i
u_1-u_2...u_i-1

(encode b_i)

a_i, b_i
u_1-u_2...u_i-1

(Say s_i = a_i + b_i = 1u_i)
```

**TASK MA**

```
START
a_i
u_1-u_2...u_i-1

(encode b_i)

a_i, b_i
u_1-u_2...u_i-1

(Say s_i = a_i + b_i = u_i or 1u_i)
```

In these addition graphs, the a_i and b_i are addends and the s_i are column sums as before. The u_i are units digits and 1 and 0, respectively, denote carry and no carry. Although 1 and 0 are actually alternative outputs, s_i = 1 u_i or 0 u_i, it is convenient to represent them as stimuli.
Notice that each processing procedure requires repeated overt responding on the part of the subject. This made it possible to both monitor progress during processing and to help insure that rehearsal was of a fixed, known variety.

The individual problems consisted of strings of the digits 0 through 9. These digits were presented orally, one at a time, as specified by the procedure in question. The strings themselves were constructed randomly, subject to four constraints: (a) no number appeared more than once in any string, (b) successive numbers (e.g., 4, 5; 7, 6) and obvious sequences (e.g., 2, 4, 6; 9, 3) were not allowed, (c) in addition tasks, as many different addend digits were employed as possible, and (d) in the MA task, carrying and noncarrying occurred randomly but equally often and with the restriction that each occur at least once.

A performance evaluation sheet was constructed for evaluating the degree to which subjects followed assigned processing procedures. The sheet contained one scale for each problem on which performance could be rated very poor, poor, fair, good, and very good.

The subjects were six volunteer graduate students at the University of Pennsylvania. They were tested individually with each task given on a different day. The order of the tasks was randomized with the constraint that each task occur once at each serial position.

Preliminary training was given on the nature of the study and on
the six tasks over a period of approximately three months prior to the experiment. During this period, the experimenter discussed various concepts related to the study such as memory load, short-term memory, information processing, rehearsal, states, and operators. The limited capacity hypothesis, that each human has a fixed finite capacity for processing information, was also described and related to the experiment. Specifically, it was pointed out that the hypothesis, if true, could be used to determine how many numbers given subjects can repeat successfully, how many they can add mentally, and so on. The exact capacity required by the various tasks was not given. The experimenter emphasized that useful data could be obtained only if subjects followed each procedure exactly as given; any deviations would make it impossible to know what the subject did and, therefore, how capacity might limit performance.

Node labelled directed graphs of the sort shown above were used to train each subject in the six processing algorithms. Subjects were instructed to concentrate on and to process each digit as an individual discreet entity. Extensive practice was provided. The conditions under which practice continued, increasingly approached those used in the experiment as a subject became more familiar with the algorithms. At first, pauses and even mistakes were commonplace but later on performance tended to become more regular. Eventually, a metronome was introduced to pace the subjects' processing. The subject and the experimenter learned to work together, speaking all digits on successive beats of the metronome. The metronome speed used varied over subjects and tasks; the speeds were chosen in order to be "comfortable" to the subject. The only restriction was that the same speed was used for all list tasks and the same speed, for all addition tasks, even though the two speeds frequently differed.

The subjects were also shown how to complete the performance evaluation record. It was emphasized that the subject was to grade his performance on each item according to how well he followed the processing algorithm, regardless of how many digits he felt he had repeated correctly. The subject was especially urged to consider such factors as how well he he concentrated on the procedure, how often and how significantly he rehearsed, whether he organized information into chunks of two or three digits, and whether he followed the algorithm exactly or added some subsidiary procedure. During this period, as well as during the experiment, the experimenter also completed a performance evaluation record, paying particular attention to those observable factors which may have
interfered with what the subject was supposed to do. These factors ranged from gross bodily movements to the subject's timing, where undue pauses in processing the numbers and the use of rhythmic patterns were of special concern.

The preliminary training and practice continued until the experimenter felt confident that the subjects were able to process the numbers automatically, under the experimental conditions and without errors or hesitations.

During the experiment itself, which was conducted within a ten day period, each subject was tested individually in a quiet room as free of distractions as possible. At the beginning of each of the six sessions, the experimenter briefly reviewed the procedure in question and emphasized that the complete cooperation and attention of the subject was needed as any deviation might seriously mar the results. He reminded the subjects that such deviations were to be reported.

Following this introduction, the subject was given five warm-up problems. Discussion of the procedure was allowed during this period. Performance was evaluated on the twenty problems which followed. Except on XL and IL problems, where the number of digits was predetermined, the experimenter continued presenting new digits on each problem until the subject made a mistake. On the XL and IL problems, the experimenter stopped presenting digits after they had been exhausted (e.g., after having read five digits of a five digit list) or after the subject had made an error. (The number of digits per problem varied from 5-9 and 4-8, respectively, with four of each type.) After each problem both the subject and the experimenter marked the performance evaluation record for that problem. Each testing session lasted approximately twenty minutes. The sessions were recorded and subject's responses were later transcribed by the experimenter.

For each item, the number of digits the subject repeated correctly at the stage immediately before he made his first error was recorded. The main dependent variable was the percentage of criterion strings of a given length that were repeated correctly (i.e., the number of times the subject repeated n digits correctly divided by the number of times the subject had an opportunity to repeat that number of digits correctly). An additional measure was taken on tasks XL and IL where list length was predetermined. The percentage of time lists of a given length were repeated correctly prior to hearing "Repeat," is directly comparable
Memory Theory in Structural Learning

to the criterion used on task L, and provided a basis for independent verification.

Results and Discussion. Application of the analytic technique to the various procedures, with respect to list length and number of digits per addend, resulted in the pattern of loads given in the table below.

<table>
<thead>
<tr>
<th>List Length / Digits per Addend</th>
<th>L</th>
<th>XL</th>
<th>IL</th>
<th>NCA</th>
<th>CA</th>
<th>MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>11</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>12</td>
<td>13</td>
<td>13</td>
</tr>
</tbody>
</table>

The maximum load is six, for example, on task L for a list of length five: one each for the individual digits and one for the list itself. On tasks XL and IL, the maximum load is also six for a list of length five before the subject is asked to "Repeat". According to our analysis, saying one, adds one to the memory load.

In task NCA, the maximum load in adding three-digit numerals is six and, as before, is attained at the second addend state: two for the first two column sums, one for the two-digit partial sum itself, one for each of the new addends, and one for the pair of addends which serves to illicit the partial sum of the third column. The maximum load in the CA task goes up to seven for three-digit numerals and occurs at the partial sum repeat state: four for the three partial sums and the numeral one, one for the four-digit chain itself, one for the stimulus u
and one for the compound stimulus consisting of the $u_j$ together with the chain. (The latter stimulus serves to determine the output $u_{j+1}$.) The maximum load in task MA depends on whether or not the last partial sum does or does not involve carrying. For a three-digit numeral, the load is seven in the former case and six in the latter.

In retrospect, the algorithms actually taught could, to some extent, have been ambiguous regarding memory load. Although only one state was specified in task L, for example, it is also possible to analyze the operation into components much as in the repeat cycles in addition. Viewed in the latter way, the load is increased by one. Task L1 was similarly ambiguous in that subjects could have held the "one" in store during the early processing. This would increase the memory load and thereby appear to lower performance to one less than capacity. The data suggested that at least one of the subjects may have done this (i.e., held the "one" in store.)

Figure 1 summarizes the experimental results adjusted for capacity and averaged over subjects for each task. Only data for those problems which were rated good or very good by both the experimenter and the subject are reported. Two of the six subjects had measured capacities of seven and four had capacities of eight as determined by their overall level of performance. The numbers indicate the percentage of problems at a given load, relative to capacity and averaged over tasks, that were responded to correctly. Note particularly that the drop at capacity (C) from 65% correct to 27% correct at capacity plus one (C+1) is more than twice that for the second largest drop of 17% between C+1 and C+2. The former

---

16 Similar analyses were conducted with all of the data with essentially the same results and are reported in Voorhies and Scandura (forthcoming).
difference, based on absolute proportion correct, is highly significant ($t_{df} = 5.35, p<.001$).

The differences between C-1 and C and C+1 and C+2 are also significant ($p<.02$ and $p<.05$ respectively), but at a lower level of significance.

Criterion performance after "Repeat" on task IL, although based on more limited data, was generally as predicted. That is, subjects with capacities seven and eight, respectively, were successful 60% of the time on lists of length five and six but only 25% of the time on lists of length six and seven. In short, the extra repetition and prefixing by "one" imposed an extra load of one.

The data for individual subjects were equally revealing. In a majority of the cases, 19 of 36 (6 tasks x 6 subjects), a sharp drop in percentage correct was observed between C and C+1. (Such "breaks" were said to occur whenever the percentage drop between successive levels was 40%+. If there was no drop of 40%, then 30-40% drops were counted.) In seven of the other 17 cases, the drop between C and C+1 was approximately as large as those between one or two other successive load levels. In only ten cases were the predicted breaks clearly in error.

<table>
<thead>
<tr>
<th>Subject</th>
<th>L</th>
<th>XL</th>
<th>IL</th>
<th>NCA</th>
<th>CA</th>
<th>MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject H</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0,+2</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>Subject J</td>
<td>-1,+2</td>
<td>0</td>
<td>-2,0</td>
<td>0,+1</td>
<td>0,+1</td>
<td>+1</td>
</tr>
<tr>
<td>Subject B</td>
<td>0</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Subject F</td>
<td>0</td>
<td>0,+1</td>
<td>0,+1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>Subject Lo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>Subject Lu</td>
<td>+1</td>
<td>0</td>
<td>-1,0,+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The above table summarizes these individual data for the respective tasks. A break at C is indicated by 0, a break at C+N is indicated by +N, and a break at C-N is indicated by -N. N equals 1 or 2 in all cases.

Although there is still obviously room for improvement, these data strongly suggest that our main objectives have been achieved. Node labelled directed graphs do seem to provide a viable basis for instructing subjects in how to process information, even on relatively complex tasks such as addition. This was possible even though the procedures actually taught are subject to further refinement.

In addition, the analytic technique proposed did provide a good fit for the data. Knowing the rule that a subject was to use, his capacity, and the problems to which the rule was to be applied, it was possible to successfully predict (ahead of time) the subject's performance on the given problems. Presumably due to idiosyncratic individual variation, which is extremely difficult if not impossible to control, there will always be perturbations in the data, but the theory seems definite enough to provide a viable means of explanation and even prediction. This is especially true if one is willing to allow statistical inference to play a central role in testing the theory, something which this author was willing to accept only reluctantly. Nonetheless, it is important to emphasize that achieving even the level of control evident in this experiment is no simple task and requires a high degree of pre-training and attention to the precise conditions under which an experiment is to be run.

Turning to our secondary objectives, the data strongly support the notion that information processing capacity has some underlying physiological base and is not subject to training. All of the subjects used the same procedures; so, it would be difficult to attribute the obtained differences in capacity to the "characteristic" use of procedures which vary in efficiency.

Furthermore, there is no a priori reason to suspect that the method used will break down with more complex tasks. Indeed, our results with the NCA task, overall, both in the experiment and in our pilot studies, seemed to be as reliable as those obtained with the simple list tasks. Why this should be so (if it truly is) is not clear, but the chances in the list tasks of unwanted chunking (via rhythm, etc.) and loss of attention (due to a less demanding task) may be correspondingly greater. A counteracting factor with more complex procedures, of course, is the amount of time required to train subjects; indeed we are not convinced
that our subjects were as highly trained, particularly on tasks CA and MA, as may be possible.

Finally, we note that rehearsal, as built into these tasks, seemed to have no real effect on performance. In the list tasks, for example, each digit was rehearsed a different number of times according to its serial position, but there was no corresponding effect on performance. These data support the notion that rehearsal, although it may provide an opportunity for recoding, does not in itself have any effect on retention.

*Further Research.* There are several major directions in which related research may go in the future. First, there is a need for more and more detailed empirical research. In the experiment reported, for example, one of the subjects performed on task IL exactly as would have been predicted if the "one" to be prefixed during the repeat stage had been kept active in short-term store during the early processing. Under these conditions, the subject's obtained performance would have been completely understandable. As it was, the apparent capacity was off by one. Experiments must be designed explicitly to determine whether such analyses are valid. To accomplish this, not only will the processing procedures have to be analyzed more rigorously, but the pre-training will undoubtedly have to be even more intensive.

Another type of experiment that needs to be run involves comparing performance on given tasks where different procedures are used. Not only would such studies deal explicitly with the question of whether or not capacity is (physiologically) innate, but they would also help determine to what extent "apparent" capacity can be modified through training. In addition, experiments of this sort could have important practical implications (e.g., consider the borrowing and equal additions methods of subtraction).

A second line of research stems from the apparent fact that there will almost certainly be some residual random error no matter how stringent the conditions under which an experiment is run. A limitation of the paradigm proposed is that during much of the processing, the processor may be operating below capacity; unknown information stored in the unused space might well interact with given processing procedures and thereby affect the results. The experimental paradigm, as described, does not allow for this possibility. In this case, one obvious alternative to experimental control is to introduce random variation into the theory. Specifically, we might assume that there are basically two ways
Structural Learning

in which this extraneous information may influence what is being processed:

1) This information may include rules which, if applied (at times when the processor is under-loaded), could serve to activate new information which replaces needed information. This would tend to increase memory load.

2) This information may serve to actually modify the given procedure, and thereby to increase the possibility of recoding (chunking) the information being processed. This would tend to decrease memory load.

Let \( \alpha \) be the probability of an extraneous element entering the processor and displacing one critical element, and let \( \beta \) be the probability of two critical elements being chunked into one unit. Then \( \alpha^n \) and \( \beta^n \), respectively, are the probabilities of \( n \) extraneous elements entering the processor and of \( n+1 \) elements being chunked as one. Given this, the probability of an error at the \( j \)th stage of processing is

\[
P(E_j) = \alpha^{C-L_j+1} \left( \sum_{i=1}^{C-L_j} \beta^i \right) + \sum_{i=1}^{C-L_j+1} \alpha^{C-L_j+1+i} \]

\[
= \alpha^{C-L_j+1} \left[ \frac{1-2\beta}{1-\beta} + \frac{\alpha\beta}{1-\alpha\beta} \right]
\]

where \( C \) is the person's capacity, \( L_j \) is the load at stage \( j \), and \( i \) is the number of critical elements recoded into one chunk. Assuming the subject never corrects himself once an error is made, the probability of making an error in a given computation is

\[
P(E) = P(E_1) + (1-P(E_1))P(E_2) + \ldots + (1-P(E_1))\ldots(1-P(E_{n-1}))P(E_n)
\]

where \( n \) is the number of stages in the computation. As is common practice in stochastic modeling, maximum likelihood estimates \( \hat{\alpha} \) and \( \hat{\beta} \) may be obtained for the parameters, and the resulting "predictions" tested.

---

\textsuperscript{17} In this derivation, we require \( \sum_{i=1}^{\infty} \alpha^i < 1 \) and \( \sum_{i=1}^{\infty} \beta^i < 1 \). These conditions insure that there is some finite probability, respectively, that none of the critical elements is displaced \( (1 - \sum_{i=1}^{\infty} \alpha^i) \) and that there is no chunking \( (1 - \sum_{i=1}^{\infty} \beta^i) \). Given this, the probability of an error at the \( j \)th stage is the sum of the probabilities of an error at that stage due to the displacement of exactly \( C-L_j+1 \) more critical elements than elements chunked. For example, \( \alpha^{C-L_j+1+2\beta^2} \) is the probability of an error due to the displacement of \( C-L_j+3 \) critical elements and chunking three elements as one.

The derivation of our formula follows directly. Let \( D_j = C-L_j+1 \).

Then

\[
\alpha^{D_j}(1 - \sum_{i=1}^{\infty} \beta^i) + \sum_{i=1}^{D_j} \beta^i \alpha^{D_j+1} = \alpha^{D_j}(1 - \beta) + \alpha^{D_j}(\beta\alpha_{1-\beta})
\]

\[
= \alpha^{D_j} \left[ \frac{1-2\beta}{1-\beta} + \frac{\alpha\beta}{1-\alpha\beta} \right]
\]
against the data. (True prediction, as opposed to explanation, would be possible to the extent that acceptable estimates can be obtained from a limited portion of the data and cross-validated against the rest.)

This is, of course, only one possible model. It is questionable, for example, whether the parameters $\alpha_j$ and $\beta_j$ for the jth stage are independent of the amount of unused space in the processor. Thus it is equally reasonable to assume that the likelihood of extraneous elements and/or chunking is a direct function of the availability of space for "turning around" on oneself so to speak. Someone not so enamored of the theory, on the other hand, might argue that extraneous elements and chunking depend on the cognitive strain involved. According to this view the closer the memory load at a stage j approaches capacity, the larger $\alpha_j$ and $\beta_j$ would be. Voorhies and I are presently investigating these possibilities.

All things considered, perhaps the most important goal of future research is to develop further implications of the general memory theory with fixed processing capacity, and to test them. Almost nothing to date has been done in this direction. Indeed, the experimental paradigm as described above is extremely limited in this regard.

In order to test most general implications of the theory, the paradigm would necessarily have to be extended to the case where more than one rule is known to be in the processor at a given time. It is not immediately clear, for example, how one might set up an experiment to test the theoretical mechanisms where a solution rule may be derived from the rules in the processor, but where the solution rule itself is not immediately available. Not only do we have to be extremely explicit in this case about the mechanisms involved (e.g., how control shifts among goals), but we must also have some way to insure that the desired rules, elements, and goals are actually in the processor and not somewhere else.

Obviously, any satisfactory solution to this problem will require a good deal of experimental trial and error together with a bit of good luck. One possible way of accomplishing this, nonetheless, might be to use pre-determined signals to activate rules, elements, and goals that have been previously learned. Once well learned and attached to suitably simple signals (e.g.,*), even relatively complex rules might be activated in this way, perhaps as easily as were the digits in our study.

Experimentation aside, there are a number of important phenomena that have been known for many years but which have defied really adequate explanation. One thing that needs to be done is to consider these
phenomena in light of the proposed theory. The Einstellung effect of Luchins (1942), for example, appears to follow directly from the theory. According to the above mechanisms, active rules are used before rules which must be derived or retrieved from long-term store, even where the latter, by most objective criteria, may be more efficient.

It is not nearly as clear, however, whether the theory provides a suitable basis for explaining why and how subjects modify known rules, sometime during the course of processing, so as to avoid going over capacity. Many subjects, for example, tend to impose a rhythm on to-be-remembered lists. This appears to improve performance, but why? Does the theory provide a suitable explanation for the phenomena? I think so, but this needs to be spelled out.

In view of our concern with more complex forms of behavior, an even more critical question involves the distinction between problem solving under the idealized conditions studied in Chapter 7, and problem solving, as it so frequently occurs in practice. Specifically, in real problem solving, people rarely finish deriving a solution procedure before parts of the procedure are actually applied. In proving mathematical theorems, for example, one frequently begins a proof before knowing for sure what steps must follow. Starting a solution, in turn, effectively changes the stimulus situation and thereby poses a reduced problem which the subject then attempts to solve. Tackling problems in this way has the major advantage of reducing cognitive strain (memory load). No one, however, has ever spelled out the exact nature of the mechanisms involved, nor shall we do so here.

Nonetheless, this phenomenon might be explained in terms of the procedures available for interpreting problem statements. Instead of interpreting a given problem statement in terms of a single goal, for example, there is no a priori reason why a subject might not interpret it alternatively as a series of subgoals. In general, people may interpret given problem statements in different ways. Accordingly, each subgoal could be dealt with in turn, presumably in accordance with the mechanisms prescribed by our theory. With this possibility in mind, I have just begun a long term, intensive study with the goal of teaching my seven-year-old son how to solve geometry construction problems. This work apparently will require training in how to construct subgoals, as this does not occur automatically. But, then, that is another story.

---

18 Recognition of this practice in human problem solving is reflected in most of the tree search methods commonly used in artificial intelligence (Nilsson, 1971), including the means-ends analyses of Simon and Newell (Volume II). In this case, checking out an operation, as a possible constituent of a solution procedure, involves actually applying the operation.
The main body of this work has been concerned with the problem of clarifying and justifying what might be called the foundations of a theory of structural learning—of making sure that the basic concepts and assumptions of the theory reflect behavioral reality. Relatively little, however, has been said about either theoretical or practical implications. This is perhaps unfortunate since my thinking has been heavily motivated not only by my continuing search for parsimony and overall unity, but also by my concern with educational problems. In concluding our study, therefore, it seems appropriate to say a few words about the relationships which exist between the theoretical edifice we have constructed and the more practical goals of education.

Rather than try to elaborate on the various brief references to education in the text, or to suggest new ones, it may be more helpful to consider in general terms a plan or conception of how the proposed theory relates to educational technology. The notion in the text of levels of theorizing is extremely basic to this conception. Also basic is Simon's (1969) conception of Sciences of the Artificial. Simon refers to natural science as knowledge about natural objects and phenomena. He then poses the question of whether there cannot also be artificial sciences—knowledge about artificial objects and phenomena. In Simon's use of the term, "artificial science" refers essentially to the science of engineering (designing, composing, synthesizing) products of one sort or another which meet certain requirements. In order to synthesize or engineer something, according to Simon, the scientist must have some purpose or goal in mind, and he must synthesize the elements at his command so as to achieve that goal while taking into account the natural laws which place constraints on the way in which these elements may operate and interact.
As an example of what Simon has in mind, consider the task of constructing a curriculum. In particular, consider the task of identifying the content and the basic processes (Scandura, 1972b) to be included in an "idealized" elementary school mathematics curriculum. In this case, the goal is to come up with an optimal curriculum—optimal in the sense that it guarantees, say, maximum transfer potential given the time limitations of a mathematics program for grades K - 6. The conceptualization which provides the constraints (natural laws) to be adhered to in achieving this goal might be the theory of knowledge described in Chapters 4, 5 and 6. The task of the curriculum engineer, then, is to devise a systematic way of achieving the goal within the constraints imposed by this theory.

An important point, one which is only implicit in what Simon says, is that the goal is what determines how much of a given theory need be taken into account in the scientific engineering of any particular product. (In my opinion, Simon puts too much emphasis on the idea that achieving goals frequently depends on only a relatively few characteristics of what he calls the "outer environment"—i.e., the conceptualization. The important point is that one must know enough about the "outer environment," or underlying theory, to achieve the particular goal in question.)

This observation provides one reason why the three partial theories proposed in the text may be particularly useful. The theory was designed (at first unconsciously so) to provide the kinds of information needed (or, equivalently, the kinds of constraints to be met) in dealing with various aspects of curriculum construction. For example, in order to identify the content and processes to be included in a curriculum, it is sufficient to consider only conceptualizations which pertain to knowledge. Other conceptual information that may be available, for example that dealing with performance and learning, may be ignored for this purpose. In fact, such information would be entirely irrelevant. On the other hand, if one wanted to deal also with the sequencing of knowledge, the assessment of what a subject knows at any given stage of learning, and/or motivation, then additional aspects of the theory would need to be taken into account. In particular, the engineering would be constrained not only by the theory of knowledge, but also by the mechanisms presumed to govern learning, performance, and/or motivation as the case may be.

Talking in terms of partial theories (levels of theorizing) is also helpful in another way. It makes explicit the well-known fact that educational development (just like development in any field) can never be entirely systematic. Although science and technology can encroach on
professional art, there will always be some residual. No matter how adequate a technology may be available at any given stage in the advance of science, there will always be certain things that need to be dealt with on intuitive grounds.

The partial theory approach makes it easier to specify and make clear the distinction between those aspects of development which relate to existing theory, and can thus be dealt with systematically (i.e., can be engineered), and those aspects of development which must be dealt with on the basis of professional know-how. Thus, given any conceptualization (e.g., a theory of knowledge), a curriculum constructor might systematically engineer the corresponding aspects of his total curriculum design (e.g., its subject matter content and processes), and then consciously (something which is rarely done) deal with those aspects of the curriculum for which the conceptualization does not provide an adequate basis (e.g., determining the entering capabilities of students).

Educational implications of the proposed theory, then, cannot be expected to follow directly from the theory itself. Such implications will depend on the extent to which the theory provides a basis for devising improved technologies for educational development and practice (i.e., new ways of manipulating the "inner environment") that are subject to the constraints imposed by one or more of the partial theories in the "outer environment."

---

1Several members of the Mathematics Education Research Group (MERG) are presently engaged in just this kind of activity. For further information, interested readers are referred to Scandura (1970a, 1971a, 1971b). Also see Scandura, An algorithmic approach to curriculum construction in mathematics, in W. E. Lamon (Ed.), Learning and the Nature of Mathematics, Chicago: Science Research Associates, 1972; and Scandura, A plan for the development of a conceptually based mathematics curriculum (In two parts), Instructional Science, 1972, in press.


Miller, G. A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 1956, 63, 81-97.


Scandura, J. M. Concept dominance in short-term memory. Journal of Verbal Learning and Verbal Behavior, 1967, 6, 461-469. (c)


Scandura, J. M. Deterministic theorizing in structural learning: Three levels of empiricism. Journal of Structural Learning, 1971, 3(1), 21-53. (b)


Scandura, J. M. A plan for the development of a conceptually based mathematics curriculum. Instructional Science, 1972, in press. (a)


Sperling, G. The information available in brief visual presentations. *Psychological Monographs*, 1960, 74, No. 11.


Additional Reference

<table>
<thead>
<tr>
<th>Author</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ackler, L.</td>
<td>113</td>
</tr>
<tr>
<td>Anderson, J. M.</td>
<td>5, 153n</td>
</tr>
<tr>
<td>Apostol, T. M.</td>
<td>166</td>
</tr>
<tr>
<td>Arbib, M. A.</td>
<td>6, 12, 37, 40</td>
</tr>
<tr>
<td>Atkinson, R. C.</td>
<td>301, 302, 303, 304</td>
</tr>
<tr>
<td>Austin, G. A.</td>
<td>175, 264</td>
</tr>
<tr>
<td>Averbach, E.</td>
<td>302n</td>
</tr>
<tr>
<td>Baker, J. C.</td>
<td>226n</td>
</tr>
<tr>
<td>Bartlett, F. C.</td>
<td>12, 113</td>
</tr>
<tr>
<td>Berlyne, D. E.</td>
<td>12</td>
</tr>
<tr>
<td>Bernbach, H. A.</td>
<td>302</td>
</tr>
<tr>
<td>Binfold, F.</td>
<td>132, 151n</td>
</tr>
<tr>
<td>Birkhoff, G.</td>
<td>44n, 166</td>
</tr>
<tr>
<td>Bobrow, D. G.</td>
<td>112</td>
</tr>
<tr>
<td>Bolzano, B.</td>
<td>166</td>
</tr>
<tr>
<td>Bourne, L. E. Jr.</td>
<td>7, 9</td>
</tr>
<tr>
<td>Bower, G. H.</td>
<td>301, 325</td>
</tr>
<tr>
<td>Bransford, J. D.</td>
<td>315</td>
</tr>
<tr>
<td>Broadbent, D. E.</td>
<td>327</td>
</tr>
<tr>
<td>Brown, E.</td>
<td>182</td>
</tr>
<tr>
<td>Brown, J.</td>
<td>326</td>
</tr>
<tr>
<td>Bruner, J. S.</td>
<td>175, 264</td>
</tr>
<tr>
<td>Chomsky, N.</td>
<td>1, 5, 40, 73n, 74, 81, 82n, 83, 84, 86, 88, 102, 113, 147, 171, 292, 316</td>
</tr>
<tr>
<td>Church, A.</td>
<td>23</td>
</tr>
<tr>
<td>Cofer, C. N.</td>
<td>141n</td>
</tr>
<tr>
<td>Conrad, R.</td>
<td>325</td>
</tr>
<tr>
<td>Corballis, M. C.</td>
<td>327n</td>
</tr>
<tr>
<td>Corcoran, J.</td>
<td>5, 67, 68, 75, 76n, 153, 153n, 154</td>
</tr>
<tr>
<td>Coriell, A. S.</td>
<td>302n</td>
</tr>
<tr>
<td>Crowder, R. G.</td>
<td>325</td>
</tr>
<tr>
<td>Dalrymple-Alford, E. C.</td>
<td>326, 327, 328</td>
</tr>
<tr>
<td>Dienes, Z. P.</td>
<td>132</td>
</tr>
<tr>
<td>Dominowski, R.</td>
<td>7</td>
</tr>
<tr>
<td>Drachman, D. A.</td>
<td>325, 332</td>
</tr>
<tr>
<td>Dunker, K.</td>
<td>113</td>
</tr>
<tr>
<td>Durnin, J. H.</td>
<td>9, 77, 79n, 81n, 104, 113n, 138, 173, 179, 182, 183n, 184n, 190, 191, 199, 200n, 212n, 219, 225, 247n, 291, 293</td>
</tr>
<tr>
<td>Ehrenpreis, W.</td>
<td>77, 81n, 113n, 212n, 247n</td>
</tr>
<tr>
<td>Einstein, A.</td>
<td>268</td>
</tr>
<tr>
<td>Ekstrand, B.</td>
<td>7</td>
</tr>
<tr>
<td>Endicott, F.</td>
<td>5, 119, 265n, 271</td>
</tr>
<tr>
<td>Engeler, E.</td>
<td>92, 145n</td>
</tr>
<tr>
<td>Ernst, G. W.</td>
<td>143, 148</td>
</tr>
<tr>
<td>Estes, W. K.</td>
<td>241, 302n</td>
</tr>
<tr>
<td>Eves, H.</td>
<td>196, 198</td>
</tr>
<tr>
<td>Feigenbaum, E. A.</td>
<td>148</td>
</tr>
<tr>
<td>Feldman, J.</td>
<td>148</td>
</tr>
<tr>
<td>Ferguson, R. L.</td>
<td>199</td>
</tr>
<tr>
<td>Fermat, P.</td>
<td>159</td>
</tr>
<tr>
<td>Finney, D. J.</td>
<td>227</td>
</tr>
<tr>
<td>Fish, D.</td>
<td>327n</td>
</tr>
<tr>
<td>Fishchof, V.</td>
<td>328</td>
</tr>
<tr>
<td>Flavell, J. H.</td>
<td>285n</td>
</tr>
<tr>
<td>Franks, J. J.</td>
<td>315</td>
</tr>
<tr>
<td>Freyd, P.</td>
<td>43, 44, 45n, 56</td>
</tr>
<tr>
<td>Furth, H.</td>
<td>285n</td>
</tr>
<tr>
<td>Gagné, R. M.</td>
<td>5, 172, 199, 212</td>
</tr>
<tr>
<td>Galanter, E.</td>
<td>5, 12, 40, 176, 269</td>
</tr>
<tr>
<td>Garstens, H. L.</td>
<td>199</td>
</tr>
<tr>
<td>Gödel, K.</td>
<td>148n, 159</td>
</tr>
<tr>
<td>Goldbach, C.</td>
<td>159</td>
</tr>
<tr>
<td>Goodnow, J. J.</td>
<td>175, 264</td>
</tr>
<tr>
<td>Greeno, J. G.</td>
<td>179, 305</td>
</tr>
<tr>
<td>Groen, G. J.</td>
<td>260, 261, 262, 263n, 264, 265, 281n</td>
</tr>
<tr>
<td>Guthrie, J. T.</td>
<td>265</td>
</tr>
<tr>
<td>Hall, M.</td>
<td>64</td>
</tr>
<tr>
<td>Hausdorff, F.</td>
<td>158</td>
</tr>
<tr>
<td>Haygood, R. C.</td>
<td>9</td>
</tr>
<tr>
<td>Henkin, L.</td>
<td>70, 148</td>
</tr>
<tr>
<td>Herstein, I. N.</td>
<td>197</td>
</tr>
<tr>
<td>Hilbert, D.</td>
<td>71</td>
</tr>
<tr>
<td>Hildebrand, D. K.</td>
<td>188n</td>
</tr>
<tr>
<td>Hill, C. E.</td>
<td>282</td>
</tr>
<tr>
<td>Hill, S. A.</td>
<td>151n</td>
</tr>
<tr>
<td>Hively, W. II</td>
<td>183n, 199</td>
</tr>
<tr>
<td>Hocutt, M.</td>
<td>88</td>
</tr>
<tr>
<td>Hovland, C. I.</td>
<td>277</td>
</tr>
<tr>
<td>Hunt, E. B.</td>
<td>277</td>
</tr>
<tr>
<td>James, W.</td>
<td>173, 325</td>
</tr>
<tr>
<td>Jeeves, M. A.</td>
<td>132</td>
</tr>
<tr>
<td>Johnstone, H. W. Jr.</td>
<td>5, 153n</td>
</tr>
</tbody>
</table>
Trabasso, T., 263, 283
Triman, D., 226n

Underwood, B. J., 239n, 241n, 264

Verplank, W. S., 259
Voorhies, D. J., 236, 263, 324n, 342n, 347

Wallach, L., 20, 21
Watkins, T., 226n
Watts, G., 226n
Waugh, D. C., 328
Weierstrass, K., 166
Wertheimer, M., 213
Whimbey, A., 328
Wickelgren, W. A., 302
Williams, J. R., 327n
Winograd, T., 146
Wittrock, M. C., 282
Witz, K., 265n
Woodward, E., 47, 137, 173, 179, 264
Woodworth, R. S., 326
Worthen, B. R., 265
Wulfeck, W. H. II, 225

Yngve, V. H., 265

Zaks, M. S., 325, 332
Zermelo, E., 158
Zorn, M., 159
SUBJECT INDEX

Algorithm, 23, 27, 28, 65
Alphabet, 22, 65, 66, 67, 89
Assessment episode, 290 def.
Assessment program, 290 def.
Association, 10, 14, 16, 17, 19, 37, 40, 81
Automaton, 27, 40
Automorphism, 54
Axiom, 62, 65-69
Axiom schema, 67
Behavior, 89 def.
  autistic, 258
  creative, 80, 134
  rule-governed, 11
Behavior class, 73
Behavior potential, 2, 41-50, 185
Behavioral objective, 111
Category, 43-46
Competence, 73, 298
  theory of, 73
Computation, 288 def.
Computing power, 74
  equivalent, 108 def.
  simple, 116
Conservation, 20, 21
Criterion referenced test, 290
Decision making capability (see also decision rule), 173
  atomic (atomic dmc), 93 def.
Decode, 17, 18, 26, 30
Description, 101
Direct product, 129
Directed graph, 34, 184
  labeled, 94
  node labeled,
Discovery learning, 229
Distinguished element, 51-55
  (D,0,R), 14, 26-32
Domain, see rule
Duality principle, 164
Einstellung, 180, 348
Embedding, 129
Embodiment, 2, 51-64
  canonical, 58
  concrete, 51
  geometric, 54
  isomorphic, 56-58
  mathematical, 51
Encode, 17, 26, 30
Environmental complex E, 287
Experience-free hypothesis, 263, 265
Expository learning, 236
Finitary conjecture, 116
Flow diagram, 34-37
Function, 12, 22
  characteristic, 65
  composite, 23
  constant, 23
  identity, 23
  recursive, 23, 24, 65
  -partial, 23
  -primitive, 24, 25
  simple, 24, 25
  successor, 23, 24, 55, 61, 124
  -inverse of, 125
  extensional study of, 25
  intensional study of, 25
Functor, 43-47
Generalization, 137
Goal, 173, 286 def.
  nth order, 293 def.
  reachable, 295 def.
  satisfied, 290 def.
Goal environment, 287 def.
Goal situation, 286 def.
  refined, 257
  resolved, 289 def.
  criterion of, 287
  -a priori, 287
  -stimulus dependent, 287
Grammar, 81
  generative, 82, 111
  linguistic, 81, 82
  transformational, 2, 6, 82, 83
  universal, 86
Group(s), 62-64
  theory of, 63, 68
Heuristic, 143, 211
Higher order hypothesis, 207
Homomorphism, 128
Human being, 176
Hypothesis testing theory, 263
366 Structural learning

Icon, 243
Image, 123
Information processing, 300
Information processing capacity, 43, 300
Innate basis, 115 def.
  finitary, 115 def.
Input array, 89 def.
Input property, 13-15
Instance, see Rule, Instance of Instantiation, 152
Instruction, 100 def.
Interpretation, 101, 237
Isomorphism, 126

Knowledge, 87
  theory of, 2, 73, 87, 105 def., 285
    -atomic, 108 def.
    -finitary, 108 def.
    -simple, 108 def.
Knowledge potential conjecture, 116

Learning, 206, 278
Learning episode, 290 def.
Limited capacity hypothesis, 327
Linguistic theory, 81

Meaning property, 22
Meaning unit, 239
Memory, 300
  active (A), 306
    theory of, 300, 320
Memory load, 189, 309
Meta-mathematics, 68
Model, 62-64
Model theory, 68-71
Modus ponens, 151
Motivation theory, 256

Output array, 89 def.
Output property, 13-15

P-marker, 82
Partition, 182, 184
  intersection, 185
    refined, 291
  rule intersection, 279
    selection intersection, 280
Path, 184, 288 def.
  equivalent, 288 def.
    simple, 288 def.
    knowledge of, 290 def.
Performance theory, 75
Potential knowledge PK, 106 def.
Predicate logic, 67, 68
Problem solving, 206
Procedure, 27, 50, 175
  deterministic, 33
  non-deterministic, 33
Production, 65-67
Program, 27, 33, 100, 101 def.
Proof, 65, 68, 157
  formal, 69
    theory of, 68-71
Quotient relationship, 130

Range, see Rule
Recursive set, 26, 65 def., 66
Recursively enumerable, 25, 65 def., 66
Reference, 237
Rehearsal, 301
Relational list, 77
Relevance class, 295 def.
Representation theorem, 64
Restriction, 137
Response(s), 10, 11
  atomic, 88, 89 def.
    effective, 13, 16
    overt, 13, 16, 18
  equivalence classes of, 13, 16
Retrieval, 308
Reversion hypothesis, 207
Rule(s), 9, 14, 28, 94 def., 176
  atomic, 94 def.
    branching, 27-32
    compatible, 214
    component, 50
    conjunction, 103 def.
      decision, 28-41, 173
    decoding, 30, 32, 34, 36, 40
      -atomic (atomic d rule), 97 def.
      degenerate, 81, 124
    elimination, 104 def.
    encoding (perceptual), 30-40
      -atomic (atomic p rule), 96 def.
      finitary, 99
    generalization, 105 def.
      -simple, 103 def.
    identity, 106
    inference, 65, 68
      -semantic, 149
      -syntactic, 151
    interpretative, 146
    inverse, 79
    n-th order generable, 107 def.
Rule (continued)
operating, 28-40
-atomic (atomic or), 92 def.
partitionally equivalent, 290 def.
relational, 77
relevant, 295 def.
restriction, 104 def.
retrieval (searching rule), 32-33
selection, 105 def., 267
simple, 101 def., 121
storing, 33
description of, 92
domain of, 92, 98
extension of, 91, 99 def.
hierarchy of, 39-41
instance of, 42, 79, 88, 90 def., 91
intension of, 91
range of, 92, 98
refinement of, 99
semantics of, 92
Rule-governed class, 15
\( \mathcal{J}: \mathcal{B} \), 90
\( \langle S, G \rangle \), 177, 207
\( \mathcal{J}: \mathcal{K} \), 91 def.
Semantics (see also Meaning), 120, 127, 238
Sign (see also Symbol), 21
minimal, 242
property of, 22
Simple element, 121
Simple performance hypothesis, 207
Statement logic, 66
State (of an automaton), 40, 323
Stimulus, 11
atomic, 88, 89 def.
effective, 13, 16
functionally distinct, 90 def.
overt, 8, 13, 18, 19, 22, 30
equivalence class of, 13, 22
Stimulus-response (S-R), 12, 15, 91
Stimulus situation, 89 def.
Stimulus trace, 18
Storage, 312
Structuralism, 4
Structural learning, idealized
(memory-free) theory of, 285, 287
Structure, 69
algebraic, 69, 70
deep, 83
Subarray, 89
Subbehavior, 90
Subprocedure, 36
Subrule, 94 def.
closed, 94, def.
Substimulus, 90 def.
Symbol (see also Sign), 22, 237, 242
Syntax, 120
System(s), 56-71
algebraic, 56, 58
formal, 65-71, 141
logical, 67
-linear, 67, 68
-suppositional, 67, 68
mathematical, 56-59
family of, 62
property of, 58-64
property schemata of, 59, 60
Theorem, 62-71
Theory, 58-64
categorical, 61, 64
deterministic, 253
formal axiomatic, 65-68
informal axiomatic, 58-64
mathematical, 63
miniature, 253
non-deterministic, 233
partial, 253
physical, 59, 62, 63
stochastic, 233
TOTE, 40, 41, 176
Training program, 290 def.
Trial, 290 def.
\( m^{th} \) stage of, 293 def.
Wff (well-formed formula), 67, 69
atomic, 69
What is learned, 180
Errata and Comments

1. p. 56 -- Condition 3 in the isomorphism definition should read
"O_1 (a_1,..., a_m) is mapped by b into O'_1 (b(a_1),..., b(a_m)) =
O'_1 (a'_1,..., a'_m) where..."

2. p. 128 -- line 3 delete "commutative"
   line 5 delete "commutative"

3. p. 148 -- footnote 20, line 7 add "to generate (proofs of) all possible...

Comments:

1. The reader should take Section 3.1 in Chapter 2, Categories and Functors, with a
grain of salt. Some time ago, when I first began sections of the book, I thought
that the formalism of categories and functors might eventually prove to be of
some use. Although authors like Gorn and Fask have apparently found more exten-
sive use for the notion, I have not (although in fairness, I should emphasize
that I have not looked very hard).

2. p. 56 -- The statement equating n-ary relations and n-l-ary operations is
imprecise. Each n-ary relation corresponds to a family of n-l-ary
operations. As Zoltan Domotor pointed out to me, relations can also
be viewed as second-order operations.

3. My use of a number of terms pertaining to mathematics is nonstandard (e.g.,
p. 62 -- equating properties with axioms; p. 63 -- partial theories are usually
called incomplete; p. 103 -- "simple generalization" corresponds to "extension";
restricting concern in Chapter 6 to morphisms (or partial morphisms) that are
computable functions). The purpose was to suggest relationships between mathe-
matical and what I believe to be psychologically relevant ideas and terminology.
The reader should not expect mathematical terms in this context to always carry
the same meaning or precision as they would in a treatise on logic or math-
ematical foundations. I hope that this will not be confusing.

4. p. 67 -- end of last paragraph before list of 5 axiom schemas -- the intent of
the sentence beginning "The set of axioms... is more easily understood
if used to start a new paragraph and read as "A finite sequence of
wffs is called a proof iff each term possesses one of the following
five forms (axiom schemas)

1. ...
2. ...
3. ...
5. ...
or else follows from a rule of inference."

* I wish to thank Zoltan Domotor for pointing out these errors and for calling my
attention to possible ambiguities.