RULES OF THE MIND

John R. Anderson
Carnegie Mellon University

With the collaboration of
Francis S. Bellezza, C. Franklin Boyle,
Frederick G. Conrad, Albert T. Corbett,
Jon M. Fincham, Donn Hoffman,
Nicholas Kushmerick, Christian Lebiere,
Mark K. Singley, and Quanfeng Wu

LEA LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
1993 Hillsdale, New Jersey Hove and London

1

Production Systems and the ACT-R Theory

John R. Anderson
Carnegie Mellon University

1.1 INTRODUCTION

The work of most cognitive psychologists is driven by the same basic question: What is happening in the human head to produce human cognition? A great frustration of our field is that as we begin to search for an answer to what seems to be a straightforward question, we discover (a) that we may not be able to find an answer to the question, (b) that we are not sure what would constitute an answer, and, indeed, (c) that we are not even sure of what the question means. The goal of this book is to describe part of the answer to that question. Given the general uncertainty of our field, however, we must first define an interpretation of that question and specify what would constitute an answer to it. These are the primary goals of this first chapter.

To avoid suspense, however, I offer here the partial answer that this book offers: Cognitive skills are realized by production rules. This is one of the most astounding and important discoveries in psychology and may provide a base around which to come to a general understanding of human cognition. I suspect, however, that most readers must be wondering what this statement really amounts to. What does it mean to say, "Cognitive skills are realized by production rules"? To help define this statement and place it in perspective this first chapter contains a brief discussion of foundational issues. Section 1.2 specifies what the theoretical status of the production system hypothesis is. Then, Section 1.3 identifies the basic features of the production rule theories of thought. Finally, Section 1.4 discusses the identifiability problems that haunt such proposals and how they are dealt with in the current approach. Each of these sections has as its goal placing the current work in proper relation to the relevant issues in cognitive science. The sections can be brief because they refer to fuller expositions of the issues elsewhere.

1.2 FRAMEWORKS, THEORIES, AND MODELS

1.2.1 Levels of Specification

Cognitive psychology (and, indeed, psychology more generally) has had an almost fatal attraction to bold, general claims about human cognition (and human nature more generally). Here are a few examples:

1. There are two memory stores: a short-term store and a long-term store.
2. Knowledge is represented in terms of visual images and words.
3. People solve problems by means–ends analysis.
4. Syntactic knowledge and general world knowledge are encapsulated in different modules.
5. Human information processing is achieved by connectionist networks of neural-like elements.
6. Cognitive skills are realized by production rules.

Each of these assertions fails the most fundamental requirement of a scientific theory: empirical falsifiability. There are ways of construing each assertion such that it could be consistent with any empirical result. For instance, almost any form of the retention function could be made compatible with the distinction between long- and short-term memory by suitable auxiliary assumptions. Yet, these assertions are transparently not without meaning and, indeed, can be elaborated into predictions that *are* empirically falsifiable. To understand what is going on here requires reviewing the distinctions among frameworks, theories, and models (Anderson, 1983a).[1]

Frameworks are composed of the bold, general claims about cognition. They are sets of constructs that define the important aspects of cognition. The distinction between long- and short-term memory, for example, would be a framework. Frameworks, however, are insufficiently specified to enable predictions to be derived from them, but they can be elaborated, by the addition of assumptions, to make them into *theories*, and it is these theories that can generate predictions. A single framework can be elaborated into many different theories. Certainly, many theories have been built around the distinction between long- and short-term memory; Atkinson and Shiffrin's (1968) theory is, perhaps, the most famous. The details that one must specify in going from a framework to a theory may seem unimportant relative to the ideas that define the framework, but they are absolutely essential to creating a true theory. For instance, it may not seem very important to the concept of short-term memory to assume it is a buffer with a fixed number of slots, but this was essential to the predictive structure of the Atkinson and Shiffrin theory.

[1]Deviating slightly from standard APA form, Anderson (without initials) refers throughout to J. R. Anderson.

Even a precise theory like Atkinson and Shiffrin's, however, is not enough to make precise predictions about a specific situation, such as a particular free recall experiment. One must make additional auxiliary assumptions to define how the theory applies to that situation. For example, within Atkinson and Shiffrin's theory, different rehearsal strategies could be assumed. The theory, with assumptions about its application to a specific situation, defines a *model* for that situation. There are many models possible within a theory, each corresponding to one way a subject could approach the situation. It is a specific model that one actually tests, although sometimes one could argue that no model derivable from the theory would be consistent with the results. It has, for example, been argued that no version of the Atkinson-Shiffrin theory could produce effects associated with depth of processing (Craik & Lockhart, 1972).

Production rules constitute a particular framework for understanding human cognition, and by now many theories have been proposed as instantiations of that framework. In 1983, I proposed a particular theory called ACT*; here I propose a variant called ACT-R.[2] The details that define a specific production-rule theory, though perhaps insignificant compared to the features that are common to defining production rules in general, are essential if a claim that "cognitive skills are realized by production rules" is to be empirically falsifiable.

1.2.2 Cognitive Architectures

Production systems are particularly grand theories of human cognition because they are cognitive architectures. *Cognitive architectures* are relatively complete proposals about the structure of human cognition. In this regard, they contrast with theories, which address only an aspect of cognition, such as those involving the distinction between long- and short-term memory. Production systems are not unique as cognitive architectures. Popular, more recent alternatives are the various connectionist theories. To go back to an earlier era, Hullian theory (Hull, 1952) would constitute a cognitive architecture, although the adjective *cognitive* might seem a little misplaced.

The term *cognitive architecture* was brought into psychology by Newell, from his work on computer architectures (Bell & Newell, 1971). Just as an architect

[2]The reader will note this is a step in the direction of parsimony. ACT* was pronounced act star. The current theory is pronounced act ar, deleting the consonant cluster *st*.

tries to provide a complete specification of a house (for a builder), so a computer or cognitive architecture tries to provide a complete specification of a system. There is a certain abstractness in the architect's specification, however, which leaves the concrete realization to the builder. So, too, there is an abstraction in a cognitive or computer architecture: One does not specify the exact neurons in a cognitive architecture, and one does not specify the exact computing elements in a computer architecture. This abstractness even holds for connectionist models that claim to be "neurally inspired." Their elements are in no way to be confused with real neurons.

The major assertion of this book—"Cognitive skills are realized by production rules"—is a general assertion about the architecture of human cognition. It is limited in its scope only insofar as cognitive skill does not encompass all of cognition. This book illustrates some of the scope of "cognitive skill." Along the way to making precise this general assertion, I define many more detailed assertions and present evidence for them.

A missing ingredient in the discussion so far is a specification of what constitutes a production system. The next section describes the concepts that define the production-system framework. The subsequent section addresses a fundamental indeterminacy that haunts such theoretical proposals. Although this indeterminacy is a problem for all cognitive architectures, this chapter focuses on its manifestation with respect to production systems.

1.3 PRODUCTION-SYSTEM ARCHITECTURE

The basic claim of the ACT-R theory is that a cognitive skill is composed of production rules. The best way to understand what this might mean is to consider a production-system model for a common skill, such as multi-column addition.

1.3.1 An Example Production System for Addition

Production rules are if–then or *condition–action* pairs. The *if*, or *condition*, part specifies the circumstance under which the rule will apply. The *then*, or *action*, part of the rule specifies what to do in that circumstance. Table 1.1 lists a set of five production rules that are sufficient to perform a certain amount of multi-column addition. These production rules are informally stated. The next chapter deals with the issue of how to formally specify these rules and with the sticky issue of what we claim is in the human head when we propose such a set of production rules. For now it is sufficient just to get a sense of how these production rules work. These production rules operate on addition problems such as:

```
 264
+716
```


The production rules are organized around a set of goals. One goal is always active at any point in time. The first production rule, NEXT-COLUMN, focuses attention on the rightmost unprocessed column and will start by choosing the ones column.

The next production to apply is PROCESS-COLUMN. It responds to the goal of adding the column digits, but there are other elements in its condition. The second clause, "d1 and d2 are in that column," retrieves the digits. Its third clause, "d3 is the sum of d1 and d2," matches the sum of those digits. In its action, it sets the subgoal of writing out d3. The clauses in the condition of a production respond to elements that are said to be in working memory. *Working memory* refers to the knowledge that the system is currently attending to.

TABLE 1.1
Production Rules for Addition*

NEXT-COLUMN
IF the goal is to solve an addition problem
and c1 is the rightmost column without an answer digit
THEN set a subgoal to write out an answer in c1

PROCESS-COLUMN
IF the goal is to write out an answer in c1
and d1 and d2 are the digits in that column
and d3 is the sum of d1 and d2
THEN set a subgoal to write out d3 in c1

WRITE-ANSWER-CARRY
IF the goal is to write out d1 in c1
and there is an unprocessed carry in c1
and d2 is the number after d1
THEN change the goal to write out d2
and mark the carry as processed

WRITE-ANSWER-LESS-THAN-TEN
IF the goal is to write out d1 in c1
and there is no unprocessed carry in c1
and d1 is less than 10
THEN write out d1
and the goal is satisfied

WRITE-ANSWER-GREATER-THAN-NINE
IF the goal is to write out d1 in c1
and there is no unprocessed carry in c1
and d1 is 10 or greater
and d2 is the ones digit of d1
THEN write out d2
and note a carry in the next column
and the goal is satisfied

*c1, d1, d2, and d3 denote variables that can take on different values for different instantiations of each production.

TABLE 1.2
Trace of Production Rules for Addition

```
>>>Cycle 1: NEXT-COLUMN
 Focusing on the next column.
>>>Cycle 2: PROCESS-COLUMN
 Adding FOUR and SIX to get TEN.
>>>Cycle 3: WRITE-ANSWER-GREATER-THAN-NINE
 Setting a carry in the next column.
 Writing out ZERO and going to the next column.
>>>Cycle 4: NEXT-COLUMN
 Focusing on the next column.
>>>Cycle 5: PROCESS-COLUMN
 Adding SIX and ONE to get SEVEN.
>>>Cycle 6: WRITE-ANSWER-CARRY
 Adding 1 for the carry to SEVEN to get EIGHT.
>>>Cycle 7: WRITE-ANSWER-LESS-THAN-TEN
 Writing out EIGHT and going to the next column.
>>>Cycle 8: NEXT-COLUMN
 Focusing on the next column.
>>>Cycle 9: PROCESS-COLUMN
 Adding TWO and SEVEN to get NINE.
>>>Cycle 10: WRITE-ANSWER-LESS-THAN-TEN
 Writing out NINE and going to the next column.
```

The production PROCESS-COLUMN illustrates the three major types of working memory elements. The first clause, "the goal is to solve an addition problem," matches a goal element in working memory. The second clause, "d1 and d2 are the digits in that column," matches part of the external problem representation. The third clause, "d3 is the sum of d1 and d2," matches a general fact from long-term memory. Often, goal information is distinguished from other working-memory information and *working memory* is used only to refer to non-goal information.

In this problem, the sum of the ones digits is 10 and so is greater than 9. The production that will fire in this situation is WRITE-ANSWER-GREATER-THAN-NINE. It sets a carry in the next column and writes out d2, which is the difference between d1 and 10. In setting a carry in the next column, the production rule is placing in working memory some information that will be used by the next production rule. Just as all the clauses on the condition side are conceived of as testing working memory, so too, all the clauses on the action side can be considered to be adding to working memory. Table 1.2 provides a trace of the production system solving the problem. The listing shows the production rules in the order that they fire and a little protocol generated by each production rule.[3]

[3]The actual production rule model is available in a file called Addition in the Examples folder on the accompanying disk.

As I demonstrate in this book, we can understand and tutor skills like multi-column addition by assuming that production rules like these are the embodiment of the skill. One sees compelling evidence for production models like the one in Table 1.1 by observing a child acquiring the skill of addition using just these sorts of rules. I discuss tutoring research that displays this power of production-rule models throughout the book.

1.3.2 Critical Features of a Production System

You should now have a sense of how production rules function. It is worth emphasizing their critical features:

- Each production rule is thought of as a *modular* piece of knowledge in that it represents a well-defined step of cognition.
- Complex cognitive processes are achieved by stringing together a sequence of such rules by appropriate setting of *goals* and other writing to *working memory*, and by reading from working memory.
- Essential to production rules are their *condition–action asymmetry*, which as seen in later chapters, is reflected in many asymmetries of human behavior.
- A final important feature of production rules is that they are *abstract* and can apply in multiple situations. Thus, the rules are not specific to adding the digits 4 and 6, for instance, but can apply to any pair of digits. This generality is achieved by use of variables in actual production-system formalism. In Table 1.1 this variable use is conveyed through terms like d1, but as shown in the next chapter, the informal specification in Table 1.1 underrepresents the variable use needed to get the correct generality for the rules.

There are a number of terms used to describe production system operation: *Pattern matching* refers to the process of determining if a production's conditions match the contents of working memory. Because multiple productions may match working memory, there arises the issue of deciding which of these will be performed. *Conflict resolution* is the term used to describe the process of determining which production rules to perform. When a production rule is performed it is said to *execute* or *fire*. The sequence of matching production rules, performing conflict resolution, and then firing a production is referred to as a *cycle*.

Corresponding to a production system is usually a computer program that actually simulates the behavior described by the production system. Writing a production-system model for a particular task usually takes the form of writing a set of production rules to perform the task. Indeed, production systems are often used as programming formalisms by people working in artificial intelligence who have no particular interest in cognitive modeling. Their status as programming languages has meant that production-system theories are precise and complete theories of particular tasks. This is a considerable virtue.

One problem with production-system theories has been that it is difficult to come to a deep understanding of a model without access to the actual running simulation, and access to other people's simulations has been hampered by a lack of access to appropriate machines and languages. This barrier has been substantially eliminated by advances in modern technology. This book comes with a disk that includes the ACT-R system and a number of the simulations described here.

1.3.3 Alternative Production Systems

Over the years, multiple production systems that instantiate the general framework have been proposed. An informative overview of these production systems can be found in Klahr, Langley, and Neches (1987). Production systems can be traced back at least to Post's (1943) proposal for rewrite systems. They also constituted an important formalism in Newell and Simon's work, which culminated with the publication of their 1972 book, *Human Problem Solving*. Their early work involved production systems as a theoretical language, without a corresponding running program. The first production system that was implemented as a computer program was one called PSG, used by Newell as the basis for his original papers on production-system models of mind (1972, 1973). Figure 1.1 (taken from Klahr et al., 1987) shows the lineage of production systems derived from this first implemented system. PSG was the inspiration for the ACTE production system, which was the basis for the cognitive theory proposed in Anderson (1976). Over the next 7 years, this evolved and matured into the ACT* production system reported in Anderson (1983a). The ACT* production system was never completely implemented as a running computer system. GRAPES (Sauers & Farrell, 1982), shown in the figure, and PUPS (Anderson & Thompson, 1989), not shown, were partial implementations of the theory relevant to the acquisition of cognitive skills. One of the advantages of this book is the computer simulation that more completely corresponds to the theoretical statements in the book.

Other lines of production systems have evolved from PSG. Particularly significant among these are the OPS production systems, which evolved out of a concern for how to do pattern matching and conflict resolution more efficiently. OPS5 (Forgy, 1981) and OPS83 (Forgy, 1984) have served as the basis for development of some expert systems in artificial intelligence. The most well known of these expert systems is R1 (McDermott, 1982), which configures computer systems. Laird, Newell, and Rosenbloom (1987) produced a dramatic new system based in OPS called Soar, and Newell (1991) advanced Soar as a unified theory of cognition. A number of comparisons to Soar appear throughout this book.

Anderson (1983a) referred to the PSG and OPS systems as *neoclassical production systems* to contrast them with the ACT production systems. Soar

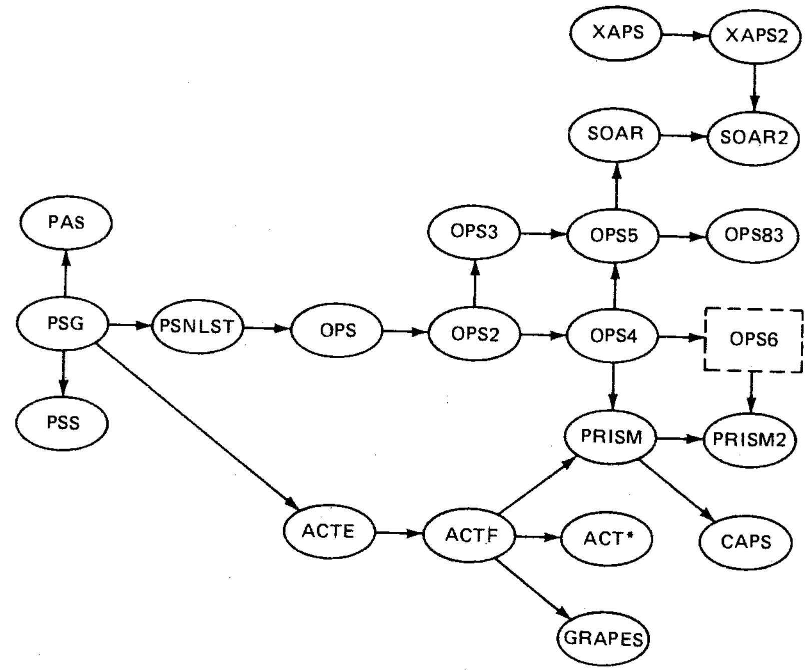

FIG. 1.1. Development of production system architectures.

certainly differs from these earlier systems in many ways and is much closer to ACT* and ACT-R, but it does preserve one important feature of the earlier systems that contrasts it with the ACT theories. This is that it represents permanent knowledge only in production rules, whereas the ACT theories propose a separate declarative representation. The next chapter discusses the significance of this procedural–declarative distinction.

1.3.4 Evidence for Production Rules

It is worth describing at the outset the general kind of evidence that indicates production rules are psychologically real, although more detailed evidence appears in later chapters. One line of evidence is simply the intuitive reasonableness of a rule set like that in Table 1.1 for describing the cognitive processes involved in a task like addition. The descriptive adequacy of a rule-based account has become apparent to most researchers in cognitive science—even those connectionists who oppose the symbol-manipulation paradigm. Thus, J. A. Anderson and Hinton (1981) acknowledged that "well-learned and regular interactions between patterns of activity can be captured as explicit rules governing manipulation of abstract symbols" (p. 31), and Smolensky (1986) recognized that "novices are described by productions with simple conditions and actions, and experts are described by complex conditions and actions" (p. 252).

Although the descriptive adequacy of such production rules has found ready acceptance, the critical question has been whether such rules are psychologically real. J. A. Anderson and Hinton went on to deny the plausibility of "models in which rules are added, deleted, or reordered" (p. 31), and Smolensky asserted that "productions are just descriptive entities" (p. 252). The critical question is what claim these rules have to psychological reality. A frequent view is that they are just descriptive approximations that obscure a deeper, underlying level where the significant psychological regularities lie. A major agenda of this book, therefore, is to show that the significant regularities in human behavior emerge at the level of production rules. It is only when a complex behavior is broken into subunits that correspond to production rules and these subunits are analyzed that we can see the detailed regularities of behavior, including the ones that connectionists are fond of. Thus, a production-rule analysis is actually critical to applying the connectionist program to complex behavior.

To establish the psychological reality of production rules, we have to commit to some theory of how production rules are executed. This is the function of the ACT-R theory that is described in the next few chapters. With that in place, I demonstrate the regularity in human performance and learning under a production-rule analysis.

In summary, the argument for the psychological reality of production rules involves two layers of evidence: One is the manifest appropriateness of rules in describing many aspects of skilled behavior. The second is the ability to predict the details of that behavior under a production-rule description.

1.4 IDENTIFIABILITY

Having explained what kind of theory ACT-R is, I come to the thorny issue of how we can know it is the correct theory. The answer might seem simple: The theory is correct if it corresponds to the available data. However, there are serious problems in using behavioral data to identify the correct theory of mind. The next subsections will explain these identifiability problems and the approach to them taken in ACT-R.

First, though, it is important to point out a tacit assumption made in the subsequent subsections. This is that the production-system framework is the right way to think about cognitive skill. The question addressed in the subsequent subsections is not how we know that there are production rules in the head, but rather, how we know ACT-R is the right production-system theory. This might well seem like we are focusing on the wrong question, but one cannot really argue for a framework because it is too poorly specified. The evidence for a framework always comes down to the success of the best theory specified within it. Thus, we have to be concerned with the details of the theory and how we know they are right. If we can get the details right, the framework will be established.

1.4.1 The Problems at the Implementation Level

For many theories, it is possible to make a distinction between an algorithm level and an implementation level (Anderson, 1987a, 1990c). That distinction is particularly well defined in the case of production systems. The *algorithm level* refers to a description of cognition in terms of the general steps of cognition. In the case of production systems, it is a description in terms of the production rules that are firing. The *implementation level* refers to a lower level description of cognition in terms of the factors that determine whether a specific production rule will fire and the speed with which it fires. The distinction is like the distinction between a high-level programming language like LISP and its machine-level implementation. Indeed, as shown in chapter 12, one can treat a production system as a programming language and simply ignore implementation issues. If we want a production system to be able to make psychological claims, however, we must be concerned with both the algorithm level and the implementation level.

As stated earlier, the details really matter when we make the claims that cognitive skills have a production-system base. Different production-system theories can differ in their details, at both the algorithm level and the implementation level. Distinguishing between different theories in terms of which production rules are actually at work (the algorithm level) is relatively unproblematic (as long as we do not get into empty debates about representation—see chapter 2). This is because the rules that are firing have such a close connection to the observed behavior. For instance, one could claim, at the algorithm level, that there was a carrying production that augmented the lower digit in an addition problem. This would be a different production system than the one in Table 1.1. It would be confirmed or disconfirmed by the observable behavior of a subject, that is, by whether the lower digit was actually augmented or not.

In contrast, there are profound difficulties in identifying what is going on at the implementation level. These difficulties arise because the theoretical claims about the implementation level are very detailed relative to the empirical data that can be used to judge the theories. These identifiability problems manifest themselves in two different ways:

1. Uniqueness. Very different proposals about what is taking place can result in the same claims about the probability and speed of production firings. These identifiability problems are rampant. For example, one might have one theory that claimed that all the production rules in Table 1.1 were matched in parallel

and another theory that claimed they were matched serially. In general, however, serial and parallel information-processing systems can be shown to be equivalent in their predictions about behavioral measures, such as processing time (Anderson, 1976; Townsend, 1974; Vosberg, 1977). Thus, behavioral data cannot distinguish between parallel and serial production matching. This is just one of the many ways that we face the fact that black boxes with very different internal structures can display identical external behavior in all respects.

2. Discovery. There is a huge space of possible implementation proposals and little guidance in finding the correct one. It sometimes seems impossible to discover one that is consistent with the data. It is like trying to find a needle in a haystack. The basic problem is that there are limitless ways to imagine the internal structure of a black box. Certainly, we have seen numerous proposals for production systems, and each has proven unsatisfactory in some regard.

One can question whether these problems really exist, and if they do whether they are peculiar to production-rule modeling. I have argued at length elsewhere (Anderson, 1976, 1987a, 1990c) that these are problems for all cognitive theorizing—not just for production systems—and it seems unnecessary to repeat the arguments. The news offered here is a way to approach these problems.

One can imagine the space of all possible cognitive theories, although infinite, as distinguishable into three ordered sets. First, there is the set of all theories. A subset of that is the set of theories consistent with the behavioral data collected so far. This subset is a tiny fraction of all theories. A much smaller subset of this subset is the set of all theories consistent with all behavioral data that could be collected. The discovery problem is that this final subset is such a tiny part of the set of theories. The uniqueness problem is that there is more than one theory in this final subset.

It needs to be emphasized that these are two independent problems. Even if we could recognize the right theory when we found it (i.e., solve the uniqueness problem), we would still have the problem of finding it. Even if we could find a theory consistent with the data (i.e., solve the discovery problem), we would face the fact that there are many equivalent theories. Thus, there are two separate problems, and they require two separate solutions. Although I cannot claim to have solved either problem completely, ACT-R reflects an approach to each problem that offers the hope of eventual solutions.

1.4.2 The Neural Approach to the Uniqueness Problem

The solution to the uniqueness problem that I have adopted is to commit to a particular style of implementation. Because cognition must be implemented in the human brain, it seems transparent that the implementation of ACT-R should be in terms of neural-like computations. Thus, the constraint used to choose

among behaviorally equivalent proposals is that the mechanisms proposed correspond to what is known about neural processing. For instance, with respect to the parallel–serial issue, we know that neural computation is highly parallel. This tells us that many processes, such as memory retrieval, have to be parallel, including the matching of production rules.

The style of neural implementation assumed in ACT* and continued in ACT-R is activation based. Declarative memory structures vary in their level of activation, and this determines their availability. Also, the rate of production-rule matching is determined by the activation levels of the declarative structures the rules match. Rules compete by inhibitory processes. A major component of learning is increasing the strength of declarative structures and production rules. Thus, when we dig below the surface of an ACT theory we find a system of computation that looks much like a connectionist system. However, in ACT, these connectionist computations at the implementation level are being used to support a system that is symbolic at the algorithm level. The computer analogy is that the primitive machine operations support the symbolic processing of LISP.

We have only partially acted on our commitment to a neural-style implementation of production systems. The activation-based computations described in subsequent chapters are only a gloss of the computations a true connectionist would want to see specified in further detail. In the last chapter, I discuss some tentative ideas about further layers of elaboration.

Note, too, that the commitment to a neural-style implementation of a production system is no guarantee of a solution to the uniqueness problem. There may well be equivalent implementations consistent with what is known about neural processing. In such cases, we have to wait for more knowledge about what neural processing is like.

1.4.3 The Rational Approach to the Discovery Problem

The problem I had with the ACT* theory and other theories (even if they were neurally based) was that they do not solve the discovery problem: finding an implementation consistent with present and future data. There are an enormous number of ways to implement production systems in a neural-like system. Why believe one is more correct than another? One can try to find a theory consistent with the available data, but what reason is there to believe it will be consistent with the next empirical phenomenon? This is exactly what happened with ACT* (Anderson, 1983a). No sooner had the theory been published than results came forth that seemed inconsistent with it. Such an occurrence is hardly unique to ACT*. Of course, one can always revise the theory slightly, to make it consistent. This is what happened in Anderson (1987b), and the same thing has happened with other theories.

An infinite number of theories that are consistent with any finite body of data will make different predictions about data yet to be collected. We need some reason to believe that when we commit to one of these theories it will hold up when the new data come in. As argued elsewhere (Anderson, 1983a, 1990a), certain factors, such as parsimony, which help us choose among theories in some sciences, do not work that well in cognitive science. Biology in general and cognition in particular do not select systems on the basis of their parsimony.

It was this discovery problem that led to the rational analysis that was described in Anderson (1990a). In line with the arguments of Marr (1982), rational analysis seeks to provide some guidance in proposing the implementation details. Rational analysis was an attempt to understand cognition, based on the thesis that cognition is adapted to the structure of the environment. Anderson (1990a) was an attempt to explore this thesis with respect to the cognitive functions of memory, categorization, casual inference, and problem solving. In each case, it was argued that cognition seemed to maximize achievement of information-processing goals within the constraint of minimizing computational costs. The effort in the 1990 book was an attempt to show optimization with minimal commitment to mechanism, but rational analysis can be used to constrain the mechanisms that implement the production-system architecture. This is what happened with respect to ACT and has resulted in the new theory, ACT-R (with the R for rational).[4] The mechanisms in ACT* were tuned and slightly changed in ACT-R to yield adaptive processing under the 1990 rational analysis.

1.5 THE REST OF THE BOOK

The theory to be proposed in this book comes from the intersection of four constraints:

1. That it be consistent with the wide variety of data deemed relevant.
2. That it be expressed as a production-system architecture, which seems to capture many salient features of the performance of cognitive skill.
3. That it be implemented in terms of neural-like processes, so that it is something that might inhabit a human brain.
4. That these processes be configured to yield optimal behavior (given the statistical structure of the environment), so that we can have additional reason to believe in their correctness.

The constraints just listed are a consequence of the concern with identifiability issues. In addition to these constraints, it is worth acknowledging two other factors that influenced the shape of the theory. The first is the legacy of ACT*;

[4]When I wrote the 1990 book, I was not sure what the relationship between rational analysis and mechanistic theory was, although I did speculate that rational analysis could be used to guide mechanistic theory.

although the theory is not exactly ACT*, its structure bears considerable resemblance to that of ACT*. The second is the commitment to providing a runnable and usable system on the disk that accompanies this book. The need to have all of these claims embodied in one system forced a high degree of clarity and consistency in the development of the theory. I am hopeful that the accompanying simulation will do much to promote scientific communication.

As indicated earlier, there is a tension between grand claims at the framework level and the details that give reality to these claims at the theory level. This book shows this tension. Much of the book presents evidence that supports the general production-rule framework and discusses how that framework can be used, but the book also specifies the details of the ACT-R theory and the evidence for those details.

The next three chapters (2 through 4) are devoted to a detailed discussion of the theoretical assumptions of the ACT-R theory. Chapters 5 and 6 are concerned with research that supports critical aspects of the ACT-R conception of skilled performance. Chapters 7 through 10 report studies concerned with the acquisition of cognitive skill. The final chapters touch on wider issues related to production systems with production rules: Chapter 11 considers issues surrounding the use of intelligent tutors based on production systems; chapter 12 discusses how to build productions in ACT-R; and the concluding chapter, 13, reviews the assumptions of the theory and the prospects for the theory's future development.