In U-Shaped Behavioral Growth, S. Strauss (Ed.)
New York: Academic Press, 1982.

3

Nonmonotone Assessment of Monotone Development: An Information Processing Analysis[1]

DAVID KLAHR

At least two premises seem to underlie a workshop organized around U-shaped behavioral growth. One premise—that the phenomenon exists—is hard to doubt, given its pervasive appearance in most of the chapters in this volume. A second, more fundamental, premise is that the analysis of U-shaped phenomena sheds "a focused light on certain important issues in developmental psychology which monotonic growth functions make difficult to unearth" (Strauss & Stavy, in press). The purpose of this chapter is to subject this premise to some skeptical scrutiny.

The chapter is divided into four main sections. To set the stage for assessing the importance of U-shaped curves to developmental psychology, I first present my view of an essential characteristic of our inquiries. The second section contains my current evaluation of the relevance of U-shaped phenomena to this enterprise—with a few analogies added for emphasis. The third section contains my reaction to some specific issues raised by Strauss and Stavy in this volume (Chapter 1) and elsewhere (Strauss & Stavy, in press), and the fourth section describes some important recent work from the computer simulation area that may contribute to developmental theory.

---

[1] Preparation of this chapter was supported in part by Grant No. NIE-G-78-0035 from the National Institute of Education and in part by Grant No. BNS77-16905 A01 from the National Science Foundation. Portions of this chapter were presented under the title "Information Processing Models of Cognitive Development" in a symposium entitled "New Approaches to Cognitive Development" at the Biennial Meetings of the Society for Research in Child Development, San Francisco, March 1979.

ISBN: 0-12-673020-2

The Uniqueness of Developmental Psychology

What is developmental psychology? The developmental psychologist observes behavior at a selected time and then again at some later date as well as what the environment was during those two periods (see Figure 3.1). We can observe inputs and outputs but nothing else. All the rest is constructed. Therefore the first important task is to select interesting behaviors (B_i) and interesting environments (E_i) in which to observe those behaviors. Having done that, we can then face the task of constructing a theory to account for the behavior observed. Two-part theories are typical in developmental psychology. One part describes the system's performance at a given time, and the other, its capacity for development at that time. I have tried to indicate that in Figure 3.1 by the large system (S_i) and the subsystem within it, which is the developmental (D_i) system. Thus, the second task of the developmental psychologist is to construct a plausible and testable model of a system that could generate the observed behavior. But the essential distinction for the developmental psychologist, the focus that sets us apart from nondevelopmentalists, is that we are interested in what the developmental mechanism is, what the D function is that operates on the entire system at a given time and transforms it into a new system at a later time. Thus the third important question, the one unique to the developmental psychologist, is: "How does this developmental mechanism work?"

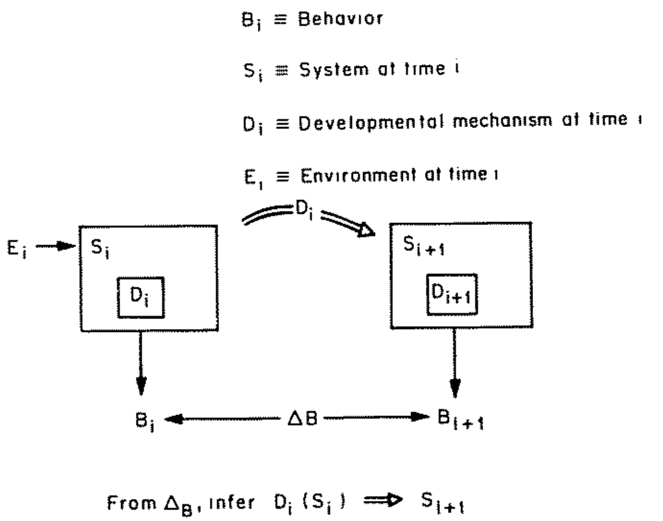

Figure 3.1. A schematic representation of the developing cognitive system. B_i = behavior; S_i = System at time i; D_i = developmental mechanism at time i; E_i = environment at time i.

Although we are all ultimately interested in the central developmental question—the question of how D_i works—not much effort is allocated to answering it. There is little theoretical development in developmental psychology. Perhaps this simply reflects the state of the art. In such a new science, the data base may not be firm enough and the phenomena stable enough, to allow for frequent theoretical advances. However, one of the things I would like to propose is that there is a line of work that we might view as the beginning of a theory of transition. Although the people engaged in this research tend not to characterize themselves as developmental psychologists, I believe that the work they are doing is of great potential relevance to the field because they construct self-modifying information processing models. Such models may be the precursors of new and important theories of cognitive development.

Later in this chapter I will return to this point to describe what it means to construct self-modifying information processing systems and what the relevance of such systems may be for developmental theory. First, I want to introduce the second strand of this chapter: the strand that relates to U-shaped curves.

Are U-Shaped Curves Important?

Implicit in Figure 3.1 is the notion that each of these systems is more sophisticated—has more knowledge and therefore leads to better performance—than its predecessor. Thus, the developmental function, the transition mechanism, would have to account for such increased performance. Now, if it is the case that in many areas we get robust decrements in performance, then it would seem that we have a challenge to any theory of self-modification: Such a theory must be able to account for the conditions under which performance decrements occur. It must describe the sort of D functions that would lead to B_{i+1} being temporarily worse than B_i. Therefore, a study of U-shaped behavioral growth might provide a vehicle, an orientation, with which to better understand and elaborate an account of how self-modification takes place.

When Sidney Strauss first invited me to a workshop on U-shaped behavioral growth, I thought it might offer an opportunity to enhance my understanding of how development takes place. However, after considering the issues and reading all the papers, I came away with a curious conclusion, one that may be a mild heresy at this conference, but one that I must state at the outset: *I believe that U-shaped curves are of no fundamental importance to developmental theory.* They may provide some interesting empirical accounts of the course of growth of some particular cognitive entity and certainly are of descriptive interest. However, I argue that they always reflect an artifact of the assessment procedure, and they must ultimately be accounted for by general mechanisms of self-modification that are neither constrained nor informed by U-shaped phenomena. That is, anybody going about building a self-modifying information processing system will have to include, in order to explain *monotone* development, all of the mechanisms that might be postulated to account for *nonmonotone* development. Therefore U-shaped curves do not provide any challenge to developmental theory.

Banking and Calculating: They Only Look U-ish

Let me start with two *non*psychological examples to illustrate why I say that U-shaped developmental curves are always measurement artifacts. For the first example, consider the organizational development of a bank. In Stage I, it is a simple bank, where everything happens under one roof. Suppose that you want to get a loan and you define the time it takes you to receive it as a measure of banking performance. At the Stage I bank you talk to the loan officer for a few hours and present your papers. He or she immediately approves the loan, but you have to allow a few days for paperwork. So, the Stage I bank gets a performance measure of 2 days (see Table 3.1).

In Stage II, the bank expands and gets more sophisticated. It acquires computers. Now there are several branch banks operating under a management policy of decentralized decision making for loans. At the Stage II bank the bank manager looks you over, says you can have your loan, the computer does all paperwork, and you come back at closing time to get your money. Your performance measure for Stage II is 1 day—a 50% improvement over Stage I.

Table 3.1
Hypothetical Performance Measures

	Stage			
	I	II	III	IV
Bank (days)	2	1	7	1/10
Calculator (key strokes)	Many	1	5	

The bank continues to grow. Now it has many more branches and a computer sufficiently powerful to centralize all decision making about loans. The branch manager in this Stage III bank is really just a customer interface now and no longer has any discretionary power. Now you go into the bank and discover that the branch manager is constrained by a central policy and a central computer. It takes you a week to get your loan, and the Stage III bank gets a very poor score with respect to your performance measure. Ultimately, this bank expands and gets more and more sophisticated. Finally, in the year 1984, the bank reaches Stage IV: To get your loan, you give it your Social Security number, your employee payroll number, your secret code, your Master Card number, your passport number, and your palm print. The computer does an instant credit check and drops the cash out in a little till at your feet. Performance for Stage IV: one-tenth of a day.

Since we are measuring the performance of this bank in terms of how long it takes to give you a loan, we discover nonmonotone behavioral growth. In fact, instead of being U-ish, this developmental curve is almost W-ish. Of course, if we had a *global* assessment of the bank's performance, we would see quite clearly that with respect to *all* its operations, it has shown monotone growth. The peaks and valleys we see are a consequence of our restricted view. Even a slight change in the narrow measure concerned with loans would have shown constant improvement. (For example, number of loans granted per week.) Moreover, the sensitivity of the loan granting decision rule has been enormously increased through the replacement of the loan officer's rules of thumb by sophisticated risk computation algorithms.

My second example uses a hand-held calculator, now common to most households. Suppose you want to measure the ease of computing a variance on a hand calculator that costs $100. Not so long ago, such a machine would have been a four-function calculator (addition, subtraction, multiplication, and division), perhaps with a square root key. Computing a variance required a lot of button pushing, which was inefficient. A few years later, a calculator in that price range had a special button: You entered a series of numbers, pushed the button, and the calculator did the variance for you. This type was often called a *statistical* calculator. Today, calculators in this price range are programmable. You can write your own program, you can save it, or you can load "canned" programs. However, the calculator no longer has a variance button. To compute a variance, you have to make several key strokes: You have to indicate that you want to load library program number 306, and *then* you'll get general purpose button A to act like a variance button. So, with respect to number of key strokes required to compute variances, the $100 calculator first showed a remarkable improvement and then a decline. Once again, by focussing on a single performance measure in a system that is undoubtedly increasing in its overall capacity and efficiency, we find a nonmonotone curve.

These examples have been chosen to illustrate the arbitrary and artifactual nature of U-shaped performance measures of the growth of complex systems. Certainly children's minds are orders of magnitude more complex than banks or calculators, and yet, with respect to the systems we are assessing, our experimental measures are often narrower than the performance measures just described. Although we may find many cases of U-shaped curves, I believe that they can tell us little about developmental processes.

Qualitative and Quantitative Differences in Knowledge Systems

Perhaps the most determined effort to demonstrate that U-shaped curves are important for developmental theory can be found in the recent work of Strauss and Stavy, reported in this volume (Chapter 1) and elsewhere (Strauss & Stavy, in press). In this section, I respond to several of their central arguments.

Extensive Quantity, Intensive Quantity, and Transformations

To solve Strauss' sugar-water problems or Siegler's conservation problems, the child must know something about transformations. Children's knowledge about the effects of such transformations must be empirically grounded, rather than inherent in some innate maturation of cognitive structures. Furthermore, they depend very heavily on the child's appropriate characterization of the effect of different types of transformations on different dimensions of the material (Strauss calls them *quantities*.) The empirical necessity follows from the fact that no transformation is *intrinsically* either preserving or changing of quantities. As shown in Table 3.2, the effect of a given transformation can be categorized only with respect to a dimension of interest.

For example, does the act of pouring conserve quantity or not? The answer depends both on what is poured and *what is measured*. If we pour a little sugar into red sugar water at 10°C, we do not change temperature, amount, height, width, or redness, but we increase sweetness. If we add more of an identical concentration, we do not change temperature, redness, or sweetness; however, the amount increases, as does liquid height, but not width (in a rigid container). On the other hand, if we add water, we increase two *extensive* quantities, reduce two *intensive* quantities, and leave one unchanged.

Table 3.2
Transformational Category for Operations on 10° C Red Sugar Water

	Pour to another container		Add material		Add more of same 10° C mix
	Same dimension	Different dimension	Sugar	Water	
Extensive dimension					
Amount	T_0[a]	T_0	T_0	T_+[b]	T_+
Height–width	T_0	$T_\pm$	T_0	T_+	T_+
Intensive dimension					
Redness	T_0	T_0	T_0	T_-	T_0
Sweetness	T_0	T_0	T_+	T_-	T_0
Temperature	T_0	T_0	T_0	T_0	T_0

[a] T_0 = null with respect to dimension (e.g. old T_p).
[b] $T_\pm$ = changes in dimension.

Our own account of the acquisition of conservation rules (Klahr & Wallace, 1976) places a very heavy emphasis on the detection of empirical regularities resulting from specific transformations of specific materials in very limited quantitative ranges. However, there is nothing inherent in a particular physical domain that is of any special psychological interest.

Quality, Quantity, Specificity, and Generality

Two intertwined dichotomies in the Strauss and Stavy chapter are qualitative versus quantitative changes in rule systems and development from specific to general, or vice versa. Their view is that the true course of development goes from general to specific, and thus a rule system must go through qualitative, rather than mere quantitative, changes. According to Strauss and Stavy, Siegler's rules for children's knowledge about the balance scale (see Figure 3.2), and, I suppose, their reformulation as production systems (see Figure 3.4; Klahr & Siegler, 1978), exhibit only quantitative change. Thus, they argue, the resultant view of development is from specific to general. This view, according to Strauss and Stavy, is incorrect.

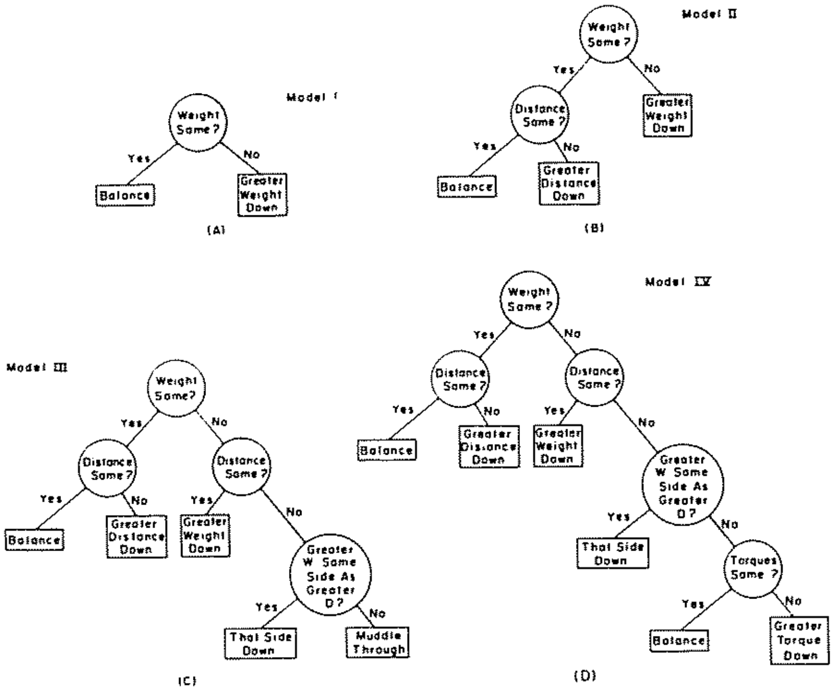

Figure 3.2. Decision tree representation for four rules about balance scales. (From Figure 1, Klahr & Siegler, 1978).

What does it mean to characterize knowledge as specific or general? In Figure 3.2, is Rule 4 more specific or more general than Rule 1? I believe that Strauss would call it more *general* because it is correct on a wider range of examples, but its conditions are more *specific*..

Siegler's rule system for balance scale predictions (Figure 3.2) exhibits several properties that are of interest. First, I think it is clear that all *tests* for the simpler rules are included in the more complex rules. Second, it is clear that the most complex rule (Rule IV), although not including a shift in dimension, does include a qualitative change: That is, to do the torque computation is to invoke a very different set of processes in addition to the simpler ones of comparing weights or distances. Also, the torque computation is configural in the sense that the effect of a given amount of weight depends on the amount of distance. Does this qualify as a qualitative change? Notice also that this system does torque computation as a last resort. If there is an easier way to make a decision, the system will make it. In that sense the earlier rules are still manifest in the more complex rules for with appropriate input Rule IV will do the same computation as Rule I.

I have been dealing with some notions that are fundamental to our understanding of developmental processes: (*a*) specificity versus generality; (*b*) inclusion versus noninclusion; (*c*) inherent contradictions; and (*d*) qualitative versus quantitative shifts. I have tried to indicate that, even after the careful attention they have received from many investigators, all of these notions can be rendered vague, imprecise, and self-contradictory. I believe that one problem lies in the medium of our theorizing. Typically, we state our developmental theories in words, with an occasional diagram thrown in for clarity. I believe that information processing models provide a vehicle for theory construction that may enable us to state much more precisely than ever before just what is going on in cognitive development.

Production Systems and Developmental Theory

For this fourth and final section of the chapter I have a two-fold goal: first, to describe a nonmodifying production system (i.e., a state description) and give an example from an area of interest to cognitive development; then to give a brief account of a few self-modifying systems. Although it would be beyond the scope of this chapter to provide a tutorial on self-modifying production systems, these brief descriptions may suffice to draw your attention to the relevant literature.

A Simple Production System

A production system consists of a set of productions. Each production—consisting of a condition and an action—has the ability to examine a data base and change that data base contingent upon what it finds there. Figure 3.3 shows a simple production system.[2] The data base has three active elements, which we can think of as the activated part of long-term memory, as the context for the current processing, and as working memory or short-term memory. This data base is examined by a set of productions presumed to exist in long-term memory. These productions are condition-action rules: They say, "If you know something about the data base, then you can add something else to the data

---

[2] Clayton Lewis first brought this simple expository technique to my attention.

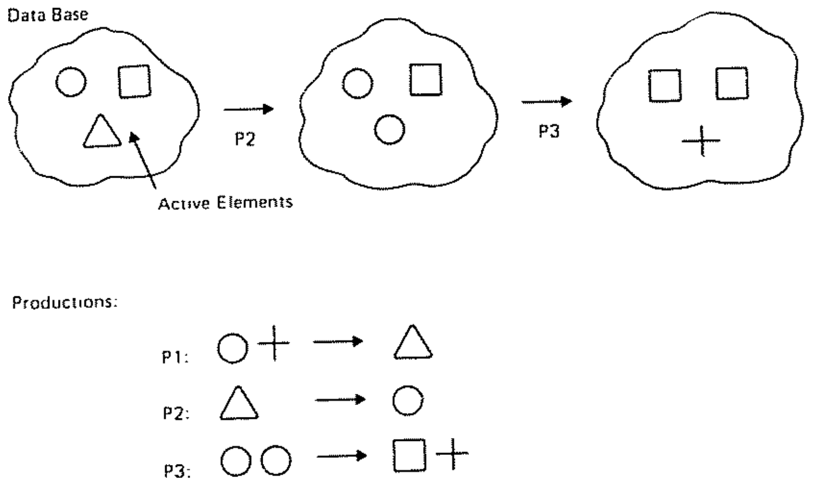

Figure 3.3. A simple production system with a data base.

base.'' The system follows a cycle of recognition and action. In this particular production system the assumption is that once a data base element matches a condition element in a production that fires, then that element is no longer available to fire any other productions unless it is reasserted into the data base. P1 says that if you have a circle and a plus, replace them with a triangle. P2 says replace a triangle with a circle; P3 says if you have two circles, replace them with a square and a plus.

If this production system were to operate on the data base shown here, it would behave as follows. On the first recognition cycle, only P2 would have all conditions matched. It would "fire," consuming its input, and add a circle to the data base. On the next cycle, neither P1 nor P2 would be able to find a complete match, but P3 would be satisfied. It would fire, effectively replacing the two circles with a square and a plus. At this point, none of the productions would be satisfied, and the system would halt.

If you take production systems seriously, you have to assume that the human information processing system contains hundreds of thousands of productions, all potentially satisfied in any cycle, but that only a limited subset of the data base is active at any one moment. Many detailed mechanisms that I cannot go into in this chapter are described in the growing literature on production systems, and the previous example should give an idea of what a production system is.

Production Systems for Balance Scale Knowledge

In terms of a state description of a particular level of performance, a production system can be written to embody a set of decision rules a subject might use to accomplish some task. For example, in a 1978 paper, Siegler and I demonstrated the logical equivalence between a set of simple binary decision trees for partial knowledge about the balance scale, shown in Figure 3.2, and a set of production systems (Figure 3.4) (Klahr & Siegler, 1978). At the level of formal analysis these two representations are equivalent. As the model of the human performance gets more complicated, of course, both representations get more complicated.

Figure 3.5 shows a decision tree for a child in a training session with the balance scale. She knows a little bit about how the balance scale works but has only qualitative encodings of weight and distance. This decision tree now begins to get complicated and it requires interpretation of information that is not explicit. The production system to do this same task consists of P1 to P8 in Figure 3.6.

This model can generate a detailed, moment-by-moment trace of the mental processes that a subject is hypothesized to use in making a prediction about which way the scale will tip, in actually seeing it tip, and then in trying to revise the hypothesis about whether weight or distance is the dominant criterion. I do not expect the reader to ap-

```
Model I
        P1:((Same W) --> (Say "balance"))
        P2:((Side X more W) --> (Say "X down"))

Model II
        P1:((Same W) --> (Say "balance"))
        P2:((Side X more W) --> (Say "X down"))
        P3:((Same W) (Side X more D) --> (Say "X down"))

Model III
        P1:((Same W) --> (Say "balance"))
        P2:((Side X more W) --> (Say "X down"))
        P3:((Same W) (Side X more D) --> (Say "X down"))
        P4:((Side X more W) (Side X less D) --> muddle through)
        P5:((Side X more W) (Side X more D) --> (Say "X down"))

Model IV
        P1:((Same W) --> (Say "balance"))
        P2:((Side X more W) --> (Say "X down"))
        P3:((Same W) (Side X more D) --> (Say "X down"))
        P4';((Side X more W) (Side X less D) --> (get Torques))
        P5:((Side X more W) (Side X more D) --> (Say "X down"))
        P6:((Same Torque) --> (Say "balance"))
        P7:((Side X more Torque) --> (say "X down"))
```

Figure 3.4. Production system (P) representations for Models I–IV. D = distance; W = weight. (From Figure 2, Klahr & Siegler, 1978.)

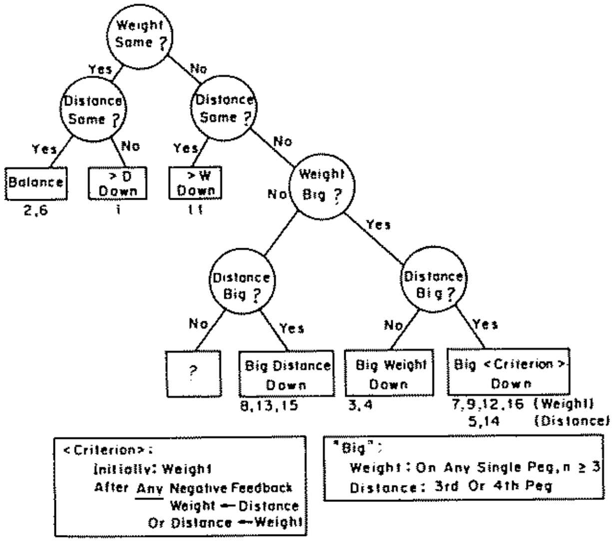

Figure 3.5. Decision tree for idiosyncratic rule used by a single child. (From Figure 3, Klahr & Siegler, 1978.)

preciate the details (that would require a careful reading of Klahr & Siegler, 1978, pp. 85–95), but I do want to convey the flavor. It is clear that to capture more of the subject's thinking processes, that is to go beyond the balance scale predictions and include more of the essential features of a training condition, we have had to increase the model's complexity.

Figure 3.7 shows a trace of the model in a Balance Scale training task. The model represents what might be in the subject's active memory after each cycle. Notice that there is quite a lot of information that the subject is presumed to know. For example, in Cycle 6, she is supposed to know:

1. She has qualitative encodings of weight and distance.
2. She is currently using a criterion that says weight is important.
3. She made a prediction based on the expectation that the left side would go down.
4. She actually saw the right side go down.
5. She compared those two.
6. She realized that she was wrong.

Now she must do something, and the next production that fires (SW2) will be one that says, "Well, if you did all that and your prediction was wrong, you must change the criterion from weight to distance."

```
<dimension.1>:(CLASS weight distance)   <dimension.2>:(CLASS weight distance)

<side.1>:(CLASS left right both)         <side.2>:(CLASS left right both)

<direction>:(CLASS up down level)


P1:((predict) (weight same) --> (made **)  (expect both level) say.b)

P2:((predict) (weight more <side.1>) --> (made **)  (expect <side.1> down) say.d)

P3:((predict) (weight same) (distance more <side.1>) --> (made **)  (expect <side.1> down) say.d)

P4:((predict) (weight more)(distance more) -->   find.big)

P5:((predict) (criterion <dimension.1>)(<dimension.1> big <side.1>)
          (<dimension.2> big <side.2>) --> (made **) (expect <side.1> down)say.d)

P6:((predict) (weight big <side.1>) --> (made **)  (expect <side.1> down) say.d)

P7:((predict) (distance big <side.1>) --> (made **)  (expect <side.1> down) say.d)

P8:((predict)(<dimension.i>) abs --> ATTEND)


E1:((expect) --> look)

E2:((expect <side.1> <direction>)(see <side.1> <direction>) --> (did **)(see ===> saw)(result correct))

E3:((expect <side.1> <direction>)(see <side.1> <direction>) abs (see) --> (did **)(see ===> saw)
                                                                         (result wrong))


SW1:((result wrong)(criterion distance) --> (old **)(distance ===> weight))

SW2:((result wrong)(criterion weight) --> (old **)(weight ===> distance))

SW3:((result correct)(criterion) --> (old **))


find.big:(OPR CALL) ;returns (weight|distance big left|right), one or two such.

look:(OPR CALL) ; looks for result of balance tipping;
                  returns (see left|right down)

attend:(OPR CALL) ; initial encoding of same or difference on distance & weight;
                    returns (weight|distance same|more left|right)
```

Figure 3.6. Production system for child during decision, feedback, and criterion revision phases of training experiment. (From Figure 4, Klahr & Siegler, 1978.)

Note that this production system deals with two kinds of knowledge that a subject uses in a task: not only the formal structure of the problem but *also the demands of the experimental situation*. This feature of production systems is of particular relevance to developmental psychology. In almost every area of cognitive development we have discovered that subtle differences in task demands may lead to widely varied performances on the part of our subjects. If we have a modeling procedure

```
      (0003|0020)
Cycle 1
 WM: ((PREDICT) (CRITERION WEIGHT))

 Fire P8: ((PREDICT) (<DIMENSION.1>) ABS --> ATTEND)
      Output from ATTEND (input to WM) ::> (weight more left)(distance more right)

Cycle 2
 WM: ((DISTANCE MORE RIGHT) (WEIGHT MORE LEFT) (PREDICT) (CRITERION WEIGHT) )
    CONFLICT.SET: (P2 P4)
 Fire P4: ((PREDICT) (WEIGHT MORE) (DISTANCE MORE) --> FIND.BIG)
      Output from FIND.BIG (input to WM) ::> (distance big right)(weight big left)

Cycle 3
 WM: ((WEIGHT BIG LEFT) (DISTANCE BIG RIGHT) (PREDICT) (WEIGHT MORE LEFT)
       (DISTANCE MORE RIGHT) (CRITERION WEIGHT))
    CONFLICT.SET: (P2 P4 P5 P6 P7)
    CONFLICT.SET: (P4 P5) AFTER SPECIAL.CASE.ORDER
    CONFLICT.SET: (P5) AFTER WM.ORDER
 Fire P5: ((PREDICT) (CRITERION <DIMENSION.1>)(<DIMENSION.1> BIG <SIDE.1>)
          (<DIMENSION.2> BIG <SIDE.2.) --> (MADE **) (EXPECT <SIDE.1> DOWN) SAY.D)

********** LEFT down

Cycle 4
 WM: ((EXPECT LEFT DOWN) (MADE (PREDICT)) (CRITERION WEIGHT) (WEIGHT BIG LEFT)
       (DISTANCE BIG RIGHT) (WEIGHT MORE LEFT) (DISTANCE MORE RIGHT))
 Fire E1: ((EXPECT) --> LOOK)
      Output from LOOK (input to WM) ::> (see right down)

Cycle 5
 WM: ((SEE RIGHT DOWN) (EXPECT LEFT DOWN) (MADE (PREDICT)) (CRITERION WEIGHT)
      (WEIGHT BIG LEFT) (DISTANCE BIG RIGHT) (WEIGHT MORE LEFT) (DISTANCE MORE RIGHT))
    CONFLICT.SET: (E1,E3)
 Fire E3: ((EXPECT <SIDE.1> <DIRECTION>)
           (SEE <SIDE.1> <DIRECTION>) ABS (SEE) --> (DID **) (SEE ===> SAW)
(RESULT WRONG))

Cycle 6
 WM: ((RESULT WRONG) (DID (EXPECT LEFT DOWN)) (SAW RIGHT DOWN) (MADE (PREDICT))
       (CRITERION WEIGHT) (WEIGHT BIG LEFT) (DISTANCE BIG RIGHT) (WEIGHT MORE LEFT)
        (DISTANCE MORE RIGHT))
 Fire SW2: ((RESULT WRONG) (CRITERION WEIGHT) --> (OLD **) (WEIGHT ===> DISTANCE))

Cycle 7
 WM: ((OLD (RESULT WRONG)) (CRITERION DISTANCE) (DID (EXPECT LEFT DOWN))
      (SAW RIGHT DOWN) (MADE (PREDICT)) (WEIGHT BIG LEFT) (DISTANCE BIG RIGHT)
       (WEIGHT MORE LEFT) (DISTANCE MORE RIGHT))
```

Figure 3.7. Trace of system shown in Figure 3.6. (From Figure 5, Klahr & Siegler, 1978.)

that accounts not just for the formal structure of the task but also for the processing requirements of the experimental situation, we might be able to resolve some of the current discussions about why versions A and B of task X lead to such wide differences in performance.

Even more important is the fact that these kinds of models give us the capability to capture the full context of training experiments—to

model some of the microstructure of the developmental process. Indeed, the model depicted in Figure 3.6 does just that; it accounts for the subject's response to negative feedback about her prediction.

Variable Condition Elements

An important feature of production systems is illustrated by productions E2 and E3 in Figure 3.6. Their purpose is to detect whether what was expected to occur actually did occur, and the feature of interest is their use of *variables* in the condition. The first element in E2—(expect ⟨side 1⟩ ⟨direction⟩)—has two variables in it: ⟨side 1⟩ and ⟨direction⟩. These are defined at the top of Figure 3.6 as small classes, *any member of which can satisfy the condition element*. Thus, if working memory (WM) contains (EXPECT LEFT DOWN) or (EXPECT RIGHT UP) etc., the *first* condition element in E2 will be satisfied. When an element is satisfied, the variable is said to be temporarily *bound* to the particular value for the rest of the attempt to match the entire condition. If WM contains (EXPECT LEFT DOWN) and (SEE LEFT DOWN), then E2 will be satisfied. More generally, E2 will be satisfied only when the system "sees" exactly what it "expects."

This ability to perform variable matches and bindings gives production systems a tremendous flexibility to vary their level of specificity, discrimination, and generalization. Variable bindings are maintained across the action side (as in P2), so that specific information detected on the condition side can be propagated, via the action side, back into working memory. This turns out to be a crucial feature of the self-modifying productions (to be described later).

Before proceeding, I must admit that there are many problems associated with the use of production systems. First of all, production systems appear very complex to people who are not familiar with them (i.e., most people, alas).[3] Second, they have many untestable assumptions built into them, and we can only decide whether the whole system makes sense, not whether any single assumption is correct. They also

---

[3] On the other hand, we include in many experimental papers a few words after each result describing the outcome of analysis of variance. Remember how many hours it took you to learn what an analysis of variance was so that "F(1,48)=9.49,p<.01" made some sense to you. Imagine what the journals would look like if we had to provide a tutorial on ANOVA to accompany each paper. Of course that is unnecessary, for we assume that the reader of psychological papers has had sufficient technical training to understand our statistical tools. This is not yet the case for those who would use information processing models. Communication is still a serious problem, and likely to remain so as the methodologies undergo rapid change and development.