COMPUTATIONAL MODELS OF SCIENTIFIC DISCOVERY AND THEORY FORMATION

Edited by
Jeff Shrager
and Pat Langley

Morgan Kaufmann Publishers, Inc.
San Mateo, California

Sponsoring Editor *Michael B. Morgan*
Production Editor *Sharon E. Montooth*
Cover Designer *Jo Jackson*
Copyeditor *Larry Olsen*
Composition *Technically Speaking Publications*
Proofreader *Martha Ghent*
Cover Mechanical *Victoria Ann Philip*

Library of Congress number: 90-4907

CIP data is available for this book.

MORGAN KAUFMANN PUBLISHERS, INC.
Editorial Office:
2929 Campus Drive
San Mateo, California
Order from:
P.O. Box 50490
Palo Alto, CA 94303-9953

Printed in the United States.

93 92 91 90 5 4 3 2 1

CHAPTER 12

Designing Good Experiments To Test Bad Hypotheses

DAVID KLAHR
KEVIN DUNBAR
ANNE L. FAY

1. Introduction

All the contributors to this volume share the ultimate goal of producing a computational model of the scientific discovery process. Thagard and Nowak (this volume) distinguish between nonpsychological and psychological approaches to this goal. The former approaches rely heavily on AI techniques, including "computational techniques different from those available to humans," whereas the latter approaches involve detailed analysis of the behavior of humans actually engaged in different aspects of scientific discovery. The psychological approaches can be further divided into two paths. One involves analyses of the scientific record of real scientists making real scientific discoveries (Darden, 1987; Kulkarni & Simon, 1988; Langley, Simon, Bradshaw, & Zytkow, 1987; Thagard & Nowak, this volume). This path is necessarily coarse-grained because the mental processes of the scientists must be inferred from historical analyses, retrospective reports, or laboratory notebooks. However, the face validity of such a data base is extremely high because it has been deliberately selected as a consequence of having produced important scientific discoveries.

The other path to a psychological understanding of the scientific discovery process—and the one we follow in this chapter—involves the cre-

ation by the analyst of simulated contexts for scientific discovery. These contexts can vary from very sparse (such as the well-known "2-4-6" task invented by Wason, 1960)[1] to highly complex (see Mynatt, Doherty, & Tweney, 1977). These simulated contexts enable researchers to perform detailed analysis of the moment-to-moment behavior of subjects as they work on a discovery task—typically ordinary college students but also scientists (Tweney & Yachanin, 1985) and young children (Dunbar & Klahr, 1989; Kuhn, Amsel, & O'Loughlin, 1988; Schauble, 1990). Although this approach has the shortcoming that the discovery task itself is only analogous to science rather than being real science, it has two important advantages. It enables us to precisely control the context of the discovery process and to obtain fine-grained, on-line observations of the thinking processes surrounding that discovery.[2]

Scientific discovery can be characterized as a process involving search in two primary problem spaces—a space of hypotheses and a space of experiments—with additional searches of subsidiary problem spaces, including an observation space, an instrumentation space, a data analysis space, and a prior literature space (Newell, 1989). In our previous work (Klahr & Dunbar, 1988), we extended Simon and Lea's (1974) dual search notion to a general framework (called SDDS for Scientific Discovery as Dual Search) for the processes that coordinate and implement dual search in the experiment and hypothesis spaces. The framework was based on our empirical observations of subjects' behavior as they formulated hypotheses and designed experiments to evaluate them. In those studies, subjects were unconstrained with respect to both hypotheses and experiments. In this chapter, we narrow our focus to ask how people search the experiment space when provided with a particular hypothesis to evaluate. By controlling both the hypothesis being

1. For recent discussions of much of the research spawned by Wason's original study, see Gorman and Carlson (1989), Klahr and Dunbar (1988), and Klayman and Ha (1987). Wason's ingenious task continues to intrigue researchers three decades after its invention (see Farris & Revlin, 1989; Gorman, 1989).

2. There is an interesting third alternative that combines the features of each of the psychological approaches. It involves the presentation of the essential knowledge context faced by major scientists at the time of their discoveries to "ordinary" but technically trained subjects to see if they can make the same discovery, given the same information and the same goals as the original discoverer (Dunbar, 1989; Qin & Simon, 1989). In the few instances in which this has been done, some subjects were able to rediscover important scientific laws.

evaluated and the extent to which it is correct, we are able to examine this search process in detail.

2. Designing Experiments

What does it take to design a good experiment? Given an hypothesis to be evaluated—either on its own merits, or in competition with alternative hypotheses—what formal rules, heuristics, and pragmatic constraints combine to yield a potentially informative experiment? How do subjects' expectations about the likelihood of an hypothesis being true or false affect the kinds of experiments that they design and their responses to information that is consistent or inconsistent with the hypothesis? In this chapter we describe a study in which we addressed these questions by presenting subjects with "bad" (incorrect) hypotheses and asking them to design a series of "good" experiments to test those hypotheses.

2.1 Previous Studies

In our earlier studies (Dunbar & Klahr, 1989; Klahr & Dunbar, 1988), we instructed subjects about all the basic features of a programmable robot and then asked them to extend that knowledge by experimentation. This training was intended to provide a rich context analogous to a scientist's partial knowledge about a domain in which further information can be obtained by experimentation. Our analyses focused on subjects' attempts to discover how a new function operates—that is, to extend their understanding about how the device works by formulating hypotheses and then designing experiments to evaluate those hypotheses; the cycle terminated when they believed that they had discovered how to predict and control the behavior of the device. To provide substantive background for the studies reported in this chapter, we start by summarizing one of our earlier studies.

2.1.1 THE BIGTRAK DEVICE

We used a computer-controlled robot tank (called *BigTrak*) that is programmed using a LOGO-like language.[3] The device is operated by press-

3. This device was first used by Shrager (1985) in his investigation of "instructionless learning" (Shrager & Klahr, 1986).

ing various command keys on a keypad. BigTrak is programmed by first clearing the memory with the CLR key and then entering a series of up to sixteen instructions, each consisting of one of its six function keys (the command) and a one- or two-digit number (the argument). When the GO key is pressed BigTrak then executes the program. To illustrate, one might press the following series of keys: CLR ↑ 5 ← 7 ↑ 3 → 15 HOLD 50 FIRE 2 ↓ 8 GO. BigTrak would then do the following: move forward five feet, rotate counterclockwise 42 degrees (corresponding to 7 minutes on an ordinary clock face), move forward 3 feet, rotate clockwise 90 degrees, pause for 5 seconds, fire twice, and backup eight feet.

2.1.2 General Method of Previous Work

First, we established a common knowledge base about the device for all subjects prior to the discovery phase. We instructed subjects about how to use each of the basic function keys. Then the discovery phase started. Subjects were told that there is a Repeat key (RPT), that it takes a numerical parameter, and that there can be only one RPT in a program. They were asked to discover how RPT works by proposing hypotheses and running programs with RPT in them to test their hypotheses.[4] Subjects generated a concurrent verbal protocol that included hypotheses, experiments (programs), observations, evaluations, and revised hypotheses.

2.1.3 Results

Nineteen of the 20 adult subjects in our first study (Klahr & Dunbar, 1988) discovered how the RPT key works within the allotted 45 minutes. The mean time to solution was 19.8 minutes. Subjects generated, on average, 18.2 programs.

Protocols were encoded in terms of the hypotheses listed in Table 1. We defined a "common hypothesis" as a fully specified hypothesis that was proposed by at least two different subjects. Across all subjects, there were eight distinct common hypotheses. Subjects did not always

4. On the original BigTrak, there was only one way that RPT worked: It repeated the previous N instructions once. In the studies reported here, we used several different rules for how RPT works.

Table 1. Common hypotheses and percentage of experiments conducted under each. Hypotheses are labeled according to the role of N: HS – Selector; HN – nil; HC – Counter.

HYPOTHESIS	CURRENT DESIGNATION*	PERCENT OF EXPERIMENTS UNDER EACH HYPOTHESIS
HS1: One repeat of last N instructions.	D	02
HS2: One repeat of first N instructions.		04
HS3: One repeat of the Nth instruction.	C	03
HN1: One repeat of entire program		06
HN2: One repeat of the last instruction		04
HC1: N repeats of entire program.	A	14
HC2: N repeats of the last instruction.	B	20
HC3: N repeats of subsequent steps.		02
Partially specified		03
Idiosyncratic		14
No Hypothesis		28

*Entries in this column show the labels for the four key hypotheses to be used in the present study Note that they include the two most "popular" (A and B) and the two least popular (D and C)

express their hypotheses in exactly this form, but there was usually little ambiguity about what the current hypothesis was. We coded each experiment in terms of the hypothesis held by the subject at the time of the experiment. Table 1 shows the proportion of all experiments that were run while an hypothesis was held. (As noted earlier, HS1 in Table 1 is the way that BigTrak actually operated.)

Subjects proposed, on average, 4.6 different hypotheses (including the correct one). Fifty-five percent of the experiments were conducted under one of the eight common hypotheses listed in Table 1. Partially specified hypotheses, which accounted for 3% of the experiments, were defined as those in which only some attributes of the common hypotheses were stated by the subject (for example, "It will repeat it N times"). An idiosyncratic hypothesis was defined as one that was generated by only one subject. Such hypotheses are not listed separately in Table 1. For 28% of the experiments, there were no stated hypotheses.

2.2 The Hypothesis Space

The eight common hypotheses—which account for over half of the experiments—can be represented in a space of "frames" (Minsky, 1975). The basic frame for discovering how RPT works is depicted at the top of Figure 1. It consists of four slots, corresponding to four key attributes:

(1) The role of *N*: does it *count* a number of repetitions, or does it *select* some segment of the program to be repeated? (2) The unit of repetition: step, program, or group of steps? (3) Number of repetitions: 1, *N*, some other function of *N*, or no role at all? (4) Boundaries of repeated segment: beginning of program, end of program, *N*th step from beginning or end? A fully instantiated frame corresponds to a fully specified hypothesis, several of which are shown in Figure 1. There are two principal subsidiary frames for RPT—*N-role:Counter* and *N-role:Selector*. Within each of these frames, hypotheses differing along only a single attribute are shown with arrows between them. All other pairs of hypotheses differ by more than one attribute. Note that the hypotheses are clustered according to the *N*-role frame in which they fall. No arrows appear between hypotheses in one group and the other because a change in *N*-role requires a simultaneous change in more than one attribute. This is because the values of some attributes are linked to the values of others. For example, if *N*-role is Counter, the number of repetitions is *N*, whereas if *N*-role is Selector, then the number of repetitions is *1*.

This frame representation is a convenient way of capturing a number of aspects of the scientific reasoning process. First, it characterizes the relative importance that subjects give to different aspects of an hypothesis. Once a particular frame is constructed, the task becomes one of filling in or verifying "slots" in that frame. The current frame will determine the relevant attributes. That is, the choice of a particular role for *N* (such as *N-role:Counter*) also determines what slots remain to be filled (such as number of repetitions:*N*), and it constrains the focus of experimentation. Furthermore, frames enable us to represent the differential importance of attributes as the "frame type" becomes the most important attribute and its "slots" become subordinate attributes. This is consistent with Klayman and Ha's (1989, p. 11) suggestion that "some features of a rule are naturally more 'salient,' that is, more prone to occur to a hypothesis tester as something to be considered." In our context, a frame is constructed according to those features of prior knowledge that are most strongly activated, such as knowledge about the device or linguistic knowledge about "repeat." When a frame is constructed, slot values are set to their default values. For example, having selected the *N*-role:*Counter* frame, values for number of repetitions, units, and boundary might be chosen so as to produce HC1 (see Figure 1).

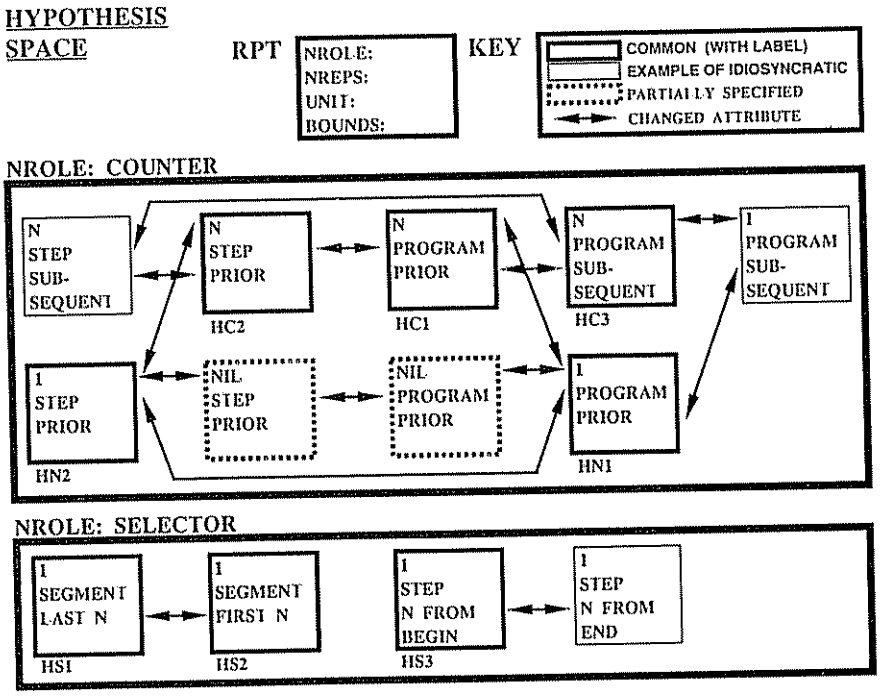

Figure 1. Frames for hypotheses about how RPT N works. Heavy borders correspond to common hypotheses from Table 1; dashed borders correspond to partially specified hypotheses; arrows indicate that adjacent hypotheses differ along a single attribute shown on the arrow; all possible hypotheses are not shown.

2.3 The Experiment Space

Subjects tested their hypotheses by conducting experiments, that is, by writing programs that included RPT and observing BigTrak's behavior. But it is not immediately obvious what constitutes a "good" or "informative" experiment. In constructing experiments, subjects are faced with a problem-solving task that parallels their effort to discover the correct hypothesis, except that in this case their search is not in a space of hypotheses but in a space of experiments.

This space can be characterized in many ways—the total number of commands in a program, the location of RPT in a program, the value of N, the specific commands in a program, the numerical arguments of specific commands, and so on. However, in this study, as in all of

our previous investigations, subjects' verbal protocols suggest that they quickly realized that there are only two key features to their experiments. The first is λ, the length of the program preceding the RPT. The second is the value of N, the argument that RPT takes. Within the λ—N space, we identify three distinct regions according to the relative values of λ and N and their limiting values.[5] Region 1 includes all programs with RPT 1. Region 2 includes all programs in which the value of N is greater than 1 but less than λ. Region 3 includes all programs in which N is equal to or greater than λ. The regions are depicted in Figure 2, together with illustrative programs from the (4,1) cell in region 1, the (3,2) cell in region 2, and the (1,4) cell in region 3.

Programs from different regions of the experiment space vary widely in how effective they are in supporting or refuting different hypotheses. Table 2 shows how BigTrak would behave under different rules when executing programs from different regions of the experiment space. The programs are depicted on the left by generic commands (e.g., X, Y, Z) and the device behavior is shown under the column corresponding to each of the four rules used in this study. To illustrate, the second example shows a two-step program with $N = 1$. This is a region 1 experiment. Under rule A, the two-step program would be executed once and then repeated one more time. Under rule B, only the last step (Y) would be repeated one additional time. Under rule C, the first step would be repeated once, and under rule D, the last N steps (in this case, the last step, since $N = 1$) would be repeated once. This program cannot discriminate between rules B and D.

The generic examples in Table 2 can represent a continuum of discriminating power. The most highly differentiated programs are obtained if we substitute distinct commands for X, Y, and Z (for example, X = ↑ 2, Y = FIRE 1). The least informative programs are those in which both the command and the parameter are the same (e.g, X = Y = Z = → 15.) Intermediate between these two extremes are programs in which the commands are the same but the parameters are different, such as FIRE 1 FIRE 2 FIRE 3. For many such programs, behavior under the different rules is in fact distinct, but it is extremely difficult to keep

5. In previous analyses (Dunbar & Klahr, 1989; Klahr & Dunbar, 1988), we used a finer-grained categorization of the Experiment Space into six regions. The mapping from the earlier to the current regions is as follows: I & II → 1, III → 2, IV & V & VI → 3. The $\lambda = 1$, $N = 2$ cell from the old region 1 goes into the new region 3.

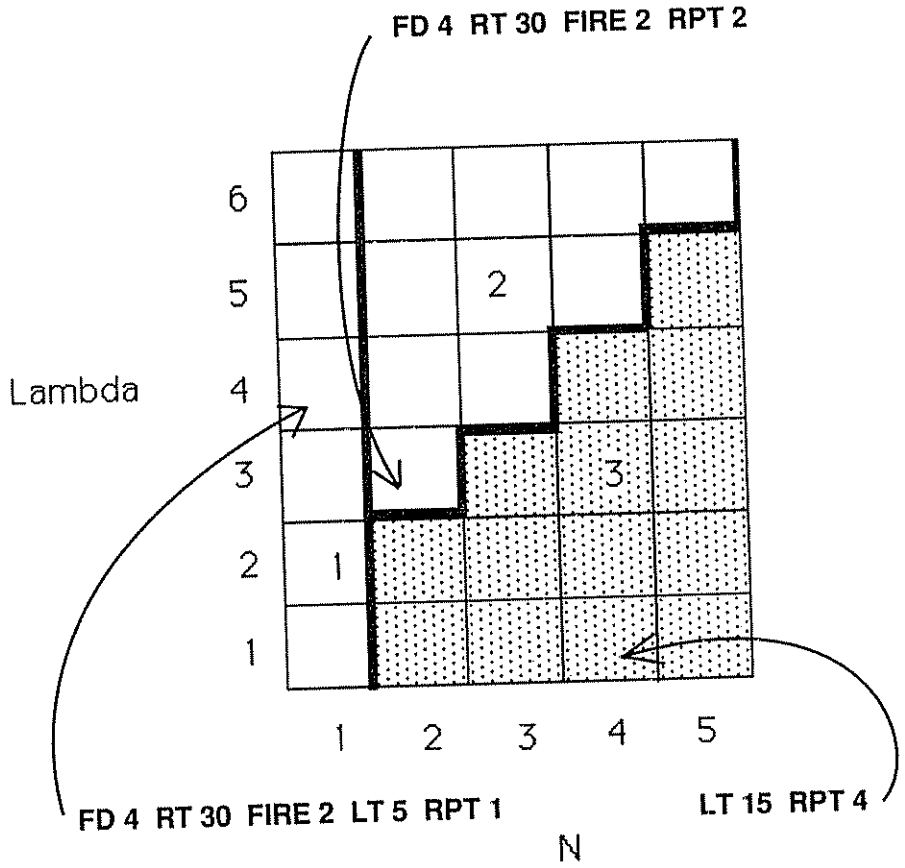

Figure 2. Regions of the experiment space, showing illustrative programs (Shown here is only the 6×5 subspace of the full 15×15 space.)

track of the BigTrak's behavior. We will present one such example in Section 4.1.

Two important and subtle features of the rules are included in the notation in Table 2. The first potentially confusing feature has to do with the ambiguity inherent in the phrase "repeat it N times." Does it mean N or $N+1$ total executions of the repeated entity? That is, if a program is supposed to repeat something twice, a subject might expect to observe either two or three occurrences of that item or segment. The underlined segments in Table 2 show the behavior generated by the $N+1$ interpretation (which is the one we use in our simulations). If subjects use the N interpretation, then they would not expect to see these extra segments. The second feature involves rules C and D when

Table 2. Behavior of BigTrak under four different rules and programs from each of the experiment space regions. Each row shows a generic program and how BigTrak would behave under each of the four rules used in this study. For each entry, executions under control of RPT are shown in boldface. (See text for further explanation.)

			COUNTERS		SELECTORS	
No.	Region	Program	A: program N times	B: last step N times	C: Nth step once	D: last N steps once
1	1	X R1	**XX**	**XX**	**XX**	**XX**
2	1	X Y R1	**XYXY**	**XYY**	**XYX**	**XYY**
3	1	X Y Z R1	**XYZXYZ**	**XYZZ**	**XYZX**	**XYZZ**
4	2	X Y Z R2	**XYZXYZXYZ**	**XYZZZ**	**XYZY**	**XYZYZ**
5	3	X Y Z R3	**XYZXYZXYZXYZ**	**XYZZZZ**	**XYZZ**	**XYZXYZ**
6	3	X Y R3	**XYXYXYXY**	**XYYYY**	**XYY***	**XYXY***
7	3	X R3	**XXXX**	**XXXX**	**XX***	**XX***
8	3	X Y Z R4	**XYZXYZXYZXYZXYZ**	**XYZZZZZ**	**XYZZ***	**XYZXYZ***

$N > \lambda$. For these programs (indicated by an asterisk), N is set equal to λ. Experiments in the three regions interact with hypotheses as follows:

1. Programs in region 1 have poor discriminating power. We have already described example 2, and example 3 similarly fails to discriminate B from D. Example 1—a minimalist program with $\lambda = N = 1$—has no discriminating power whatsoever.
2. Region 2 provides maximal information about all the most common hypotheses because it can distinguish between Counters and Selectors, and it can distinguish *which* Selector or Counter is operative. It produces different behavior under all four rules for any program in the region, and varying N in a series of experiments in this region always produces different outcomes.
3. Under rules C and D, results of experiments in region 3 may be confusing because they are executed under the subtle additional rule that values of N greater than λ are truncated to $N = \lambda$. Therefore, varying N in this region will give the impression that N has no effect on the behavior of the device (compare examples 1 and 7 or 5 and 8). Although some of the programs in this region are discriminating (such as example 5, with $\lambda = N = 3$), others either do not discriminate at all (C versus D in example 7) or depend on the truncation assumption to be fully understood (such as examples 6 to 8).

3. An Empirical Study of Testing an Incorrect Hypothesis

The brief summary of our earlier studies provides a context for the following description of a new study designed to explore subjects' responses to negative feedback. In these new studies, we always provided subjects with an initial hypothesis about how RPT might work. It was always wrong. In some conditions, it was only "somewhat" wrong, in that it was from the same frame as the way that RPT actually works. In others, it was "very" wrong, in that the suggested hypothesis came from a different frame than the actual rule.

3.1 Subjects

Thirty-six Carnegie Mellon undergraduates (27 males and 9 females) participated in the experiment for course credit. Most were science or engineering majors. All the subjects completed a questionnaire about

their programming experience and skill and a self-rating of their skill in math, science, and mechanical reasoning. Subjects reported having taken between zero and five programming courses (mean 2.1, sd 1.4), and they tended to rate themselves between average and above average on all the technical and scientific scales.

3.2 Procedure

All subjects worked with a simulated version of the BigTrak on a Xerox Dandelion workstation. The simulator included five command keys (↑, ↓, ←, →, and FIRE). The workstation enabled us to control the way that RPT actually functioned and facilitated the recording of the programs that subjects wrote to evaluate their hypotheses.

There were three phases to the study. In the first, subjects were introduced to the (simulated) BigTrak and were trained to criterion on all its basic commands. In the second phase, subjects were told that there was a RPT key, that it required a numeric parameter, and that there could be only one RPT in a program. They were told that their task was to find out how RPT worked by writing programs to test a particular hypothesis. At this point, the experimenter suggested one possible way that RPT might work and instructed the subject as follows:

> Write down three good programs that will allow you to see if the Repeat key really does work this way. Think carefully about your program, and then write the program down on the sheet of paper... Once you have written your program down, I will type it in for you, and then I will run it. You can observe what happens, and then you can write down your next program. So you write down a program, then I will type it in, and then you will watch what the program does. I want you to write three programs in this way.

Next, the third phase began. Subjects wrote programs (experiments) to evaluate the given hypothesis. Although subjects did not have access to a record of the behavior of the device in earlier experiments, they did have access to the list of programs that they had written, and they often referred to them in commenting on differences among the most recent outcome and previous ones. Subjects were instructed to give verbal protocols. This gave us a record of (1) what they thought about

Table 3. Percentage of experiments run in previous studies for each Repeat rule used in the present study.

	STUDY 1	STUDY 2
COUNTERS		
A: REPEAT THE ENTIRE PROGRAM *N* TIMES	14	13
B: REPEAT THE LAST STEP *N* TIMES	20	26
SELECTORS		
C: REPEAT THE *N*TH STEP ONCE	3	5
D: REPEAT THE LAST *N* STEPS ONCE	2	5

the kinds of programs they were writing while testing their hypotheses, (2) what they observed and inferred from the device's behavior, and (3) what their hypothesis was about how RPT actually worked. When subjects had written, run, and evaluated three experiments, they were given the option of writing additional experiments if they were still uncertain about how RPT worked.

3.3 Design

The BigTrak simulator was programmed so that each subject worked with a RPT command obeying one of the four rules listed in Table 3. Note that there are two Counter rules and two Selector rules. Table 3 also summarizes the results from two earlier studies showing the relative frequency with which subjects ran experiments to test four specific hypotheses. Two of them (A and B) accounted for approximately one-third of all experiments, while the other two (C and D) were tested on only 10% of all experiments. More generally, all of our earlier studies showed that Counter hypotheses were regarded as highly probable and Selector hypotheses were regarded as improbable. This consistent preference made it possible for us to investigate the extent to which the a priori belief in particular hypotheses affected the types of experiments designed and the interpretation of results.

The key feature of this study is that *RPT never worked in the way that was suggested.* The design is shown in Table 4. The Given hypothesis

Table 4. Design of given-actual conditions.

	ACTUAL	
GIVEN	COUNTER	SELECTOR
COUNTER	A → B B → A	A → C A → D B → C B → D
SELECTOR	C → A C → B D → A D → B	C → D D → C

is the one that was suggested by the experimenter, and the Actual hypothesis is the way that BigTrak was programmed to work.[6] We used a between-subjects design, with three subjects in each of the given-actual conditions ($N = 36$). This yielded 12 subjects in within-frame conditions; for example, Repeat the program N times (A) → Repeat the last step N times (B); Repeat last N steps once (D) → Repeat Nth step once (C). There were 24 subjects in the between-frame conditions; for example, Repeat the program N times (A) → Repeat last N steps once (D); Repeat Nth step once (C) → Repeat the program N times (A).

3.4 Questions About Searching the Experiment Space

Now that we have described the details of our design, we can pose some specific questions. First, with respect to overall effort and success rates:

1. Will subjects have more difficulty when they have to change frames in order to discover the Actual rule than when they can remain within the same frame as the Given hypothesis?
2. Will subjects find it easier to discover rules from the preferred frame (Counters) than from the nonpreferred frame (Selectors)?
3. Will the difficulty of crossing frame boundaries interact with preferences for hypotheses? That is, will it be easier to discover a Counter

6. In our discussion, we will distinguish among three categories of hypotheses and rules. *Given* hypotheses are the ones initially suggested by the experimenter, *Current* hypotheses are the ones currently being evaluated by the subject, and *Actual* rules are the ways that RPT actually works in a particular condition. Ideally, a subject would start with Current = Given and end with Current = Actual.

when given a Selector hypothesis than to discover a Selector when given a Counter hypothesis?

4. Will the extent of "rational" search of the hypothesis space depend on experimental conditions? To what extent will subjects propose hypotheses that are consistent with the evidence available to them?

5. Will subjects find it easier to reject hypotheses that have been given to them rather than hypotheses that they have generated themselves? In our prior research, most subjects began with hypotheses A and B and needed a considerable amount of disconfirming evidence before abandoning their hypotheses. When subjects are actually given hypotheses to test, they may more readily abandon their hypotheses.

Second, with respect to search in the experiment space:

1. Will subjects' interpretation of what a "good experiment" is vary according to the Given-Actual condition in which they find themselves? That is, will experiments for favored hypotheses tend to demonstrate the presumed effect of RPT and experiments for unfavored hypothesis tend to have the power to discriminate between alternative hypotheses?

2. To what extent will subjects adopt the same experiment space that we have presented here? Will their choice of experiments reflect an implicit understanding of the interactions shown in Table 2?

3. What kinds of pragmatic rules will subjects apply to their search of the experiment space? Will they design programs that are easily observable, discriminatory, and memorable?

Finally, with respect to the observation and encoding of experimental outcomes:

1. Given that the BigTrak never works the same way as the Given hypothesis, how will subjects interpret the disconfirming evidence?

2. Will disconfirmation result in subjects searching a new region of the experiment space?

3. Will hypothesis preference also influence subjects' encoding and evaluation of experimental outcomes as well as overall success rates? That is, will subjects tend to distort their encoding of evidence in the direction of confirming favored hypotheses?

4. Experimental Results

The raw data are comprised of subjects' written programs as well as transcriptions of subjects' protocols (verbalizations) during the experimental phase. The protocols provided the basis for all of our measures of hypotheses changes and search in the experiment space. In Section 4.1, we informally describe two characteristic protocols, and then in subsequent sections we provide a quantitative analysis based on the full set of protocols.

4.1 Complete Protocols

The subject protocols are extremely rich, and in this section our aim is only to convey a general sense of the kind of encodings and inferences that we make from them. In the following two summaries, we focus on the ease with which subjects coordinate their search in the hypothesis and experiment spaces. The complete protocols are listed in Appendix A and Appendix B. Line numbers correspond roughly to major clauses. For each experiment, the commands used in the program are on the left side of the listing, and the actual behavior of BigTrak is shown in boldface type on the right side. Experimenter comments are shown in uppercase.

4.1.1 SUBJECT DP

Subject DP had experience with several programming languages (LOGO, LISP, PASCAL) and reported between 100 and 500 hours of programming experience. He rated himself as "above average" in mathematics and science, and average in "handling new gadgets." DP was in the Counter → Selector condition; he was given rule A: *Repeat entire program N times.* The actual rule was rule C: *Repeat Nth step once.* DP discovered the correct rule after five experiments.

Several general characteristics of DP's protocol make it interesting (but not unusual). First, even before the first experiment, DP rejected the given hypothesis and proposed an alternative (003: "I want to test to see if Repeat repeats the statement before it"; for example, this is rule B, not rule A.) Second, throughout the experimental phase, DP made many explicit comments about the attributes of the experiment space. He clearly attended to the properties of a "good" experiment.

Third, DP operated in an experiment space that included a feature that we have ignored so far—whether the range of influence of RPT extends to commands that precede it, follow it, or both. (We have included only the first of these in our analysis so far.) Several of our subjects explored this possibility, but it was not a dominant focus for most experiments.

DP first focused on the question of the before/after range of RPT, and he wrote a minimal program with one step on each side of RPT. Note that he used easily discriminated commands (left and right turns) so that, if RPT was having an effect on either side of its location in the program, it would be unambiguously evident. (This ability to write programs that contain useful "markers" is an important feature of our subjects' behavior, and we will return to it later.) DP was very clear about his intentions in his first experiment (003–010): to determine whether RPT acts on instructions before or after the RPT command. To resolve this question, DP conducted an experiment with commands both before and after the RPT key. This experiment was appropriate as it allowed DP to discriminate between these two rival hypotheses. However, with respect to being able to discriminate between the Given hypothesis (A), the Current hypothesis (B), and the Actual hypothesis (C), the program yields ambiguous results. DP extracted from the first experiment the information he sought (017): "It appears that the Repeat doesn't have any effect on any statements that come after it."

For the second experiment DP returned to the question of whether the Given hypothesis (A) or the Current hypothesis (B) was correct, and he decided to increase λ from 1 to 2. He also decided to include one step following the RPT "just to check" that RPT had no effect on instructions that followed it (022-023). Thus, DP was in fact testing three hypotheses; A, B, and "after." Once again, he used commands that could be easily discriminated. He continued to write a program from region 3 of the experiment space ($\lambda = 2$, $N = 2$). DP observed that there were two executions of the ↑ 2 instruction, and he concluded (028) that "it only repeats the statement immediately in front of it." This conclusion is consistent with the data that DP had collected so far, but the hypothesis (B) was not in fact how the RPT key worked.

For the third experiment, DP continued to put commands after RPT just to be sure they were not affected. However, given that his Current hypothesis was confirmed in the previous experiment, he next decided to write a program that further increases the length of the program. This

was his first experiment in region 2. The goal of this experiment was to "see what statements are repeated" (032). He realized that the outcome of this experiment was inconsistent with his Current hypothesis (B), whereas the outcome of the previous experiment was consistent with B (050): "It seemed to act differently in number 2 and number 3." The unexpected result led DP to abandon hypothesis B, and he decided to continue beyond the mandatory three experiments.

For the fourth experiment, DP used a different value of N (055): "just to see if that (a value of 3 instead of 2) has anything to do with it." Here, too, DP demonstrated another important characteristic of many of our subjects' general approach to experimentation. He used a very conservative incremental strategy, similar to the VOTAT (vary one thing at a time) experimental strategies described by Tschirgi (1980) and the Conservative Focusing strategy described by Bruner, Goodnow, and Austin (1956). This approach still led him to put commands after the RPT, even though he seemed confident that RPT had no effect on them and even though they placed greater demands on his observational and recall processes. At the $\lambda - N$ level, DP executed VOTAT consistently throughout his series of five experiments. The $\lambda - N$ pairs are: 1–2, 2–2, 3–2, 3–3, 3–1. For the last three experiments, even the specific commands and their parameters remained the same, and only N varied. This moved him from region 2 into region 3. While analyzing the results of this experiment (061–071) in conjunction with earlier results, DP changed from the Counter frame to the Selector frame. First he noticed that "the number three" statement (the ↓ 1) was repeated twice in this case but that "the turning statement" was repeated (executed) only once (061–063). The implied comparison was with the previous experiment, in which the turning statement ("the right 15 command" [064]) was the command that got repeated. The next sentence is of particular interest:

> ...because when I change the number not only did it change ...it didn't change the uh ...the number that it repeated but it changed the uh... the actual instruction (066–069).

We believe that DP was attempting to articulate a change from the Counter frame to the Selector frame, as the following paraphrase of his comments indicates:

> When I changed the value of N, it didn't change the *number* of repetitions, but it did change *which* commands got repeated.

DP went on to clearly state two instantiated versions of the correct rule by referring to previous results with $N = 2$ and $N = 3$, and he designed his fifth experiment to test his prediction with $N = 1$. The outcome of this final experiment, from region 1, in conjunction with earlier results, was sufficient to convince him that he knew how RPT worked.

4.1.2 SUBJECT JS

JS rated himself as above average in mathematics and science as well as in "handling new gadgets." He reported having between 50 and 100 hours of programming experience. This subject was also in the A $\rightarrow$ C condition. JS's protocol had two interesting features. First, he never fully accepted the Given hypotheses (A: Repeat entire program N times), and at the very outset he proposed a few alternatives. Second, he was very articulate about several aspects of his experimental strategy, not only with respect to both the $\lambda - N$ space but also in terms of the logic of a disconfirming strategy and pragmatic constraints, such as designing programs that are easy to observe and encode.

JS started by expressing doubt about the Given hypothesis and setting out to disconfirm it (002–005) using a "simple program" (006) with "distinct steps" (009) that could be "distinguished" (012). As he developed the program, he proposed two alternative hypotheses and reasoned about them on the basis of plausibility and functionality (013–017). As he developed his first program, JS described its predicted behavior as if his Current hypothesis were Repeat Nth step once, which was the Actual rule. That is, he expected the RPT 1 to execute the $\uparrow$ 1 after the $\downarrow$ 1, which would "bring it back to its original position" (022). JS also added a command following the RPT just to see if RPT had any effect on subsequent commands, although he did not seem to expect it.

The experimenter now asked JS to make a prediction before running the program (032), and JS gave two possible outcomes. He predicted that, if the Given hypothesis was correct, then, after the program was executed the first time, it would be executed again in its entirety: "It will continue with the rest of the program" (037). However, if his alternative hypothesis (C) was correct, then "the only thing I'm thinking it might do is I think it might just move forward 1 (repeat the first step only), and then it'll end up turning to the left 30" (038–040).

The program ran, and JS correctly observed and interpreted its behavior as disconfirming the Given but confirming his Current (and the Actual) hypothesis (042–049). However, JS then realized that a region 1 program did not rule out another plausible hypothesis: Repeat first N steps once. He deliberated for a bit on what kind of experiment would best discriminate between the two possibilities, and for his second experiment he constructed a region 2 program with $\lambda = 4$ and $N = 3$ and four highly discriminable commands. He also articulated a VOTAT strategy (061–063): "I want to run the same program because I know what it does. I just want to change the condition of the repeat."

At this point, JS stated the correct rule as well as his now-disconfirmed hypothesis (007–082):

> So it's just repeating the step number of the...the number you put after the Repeat it repeats that sequence,...it doesn't repeat first, second, and third like I thought it might, it just repeats the third step.

Having discovered the correct rule, JS went on to explore the effect of having $N > \lambda$ and wrote one more experiment to attempt to resolve that question. He appeared to end somewhat unsure of this subtle feature.

4.1.3 General Features of the Subjects' Behavior

We have presented only two of our 36 protocols, but they suffice to illustrate several general features of the subjects' approach to this task. First, subjects did not always accept the Given hypotheses, even before running their first experiment. (Recall that both JS and DP expressed doubt about the Given hypothesis *prior* to running their first experiment and proposed an alternative.) This initial skepticism varied in its degree and in the conditions under which it occurred. We can define "mild skepticism" as the consideration of an alternative hypothesis from the same frame as the Given hypothesis and "extreme skepticism" as the consideration of an alternative from a different frame. For both Counter and Selector subjects, nearly two-thirds (17/26) of the hypotheses considered in experiment 1 (for all 18 subjects in each condition) were, in fact, the Given hypotheses. However, which *additional* hypotheses were generated by subjects depended on the Given condition. Whereas 78% of all non-Given hypotheses suggested by Selector subjects were Counters, only 22% of all non-Given hypotheses suggested by Counter

subjects were Selectors. Extreme skepticism occurred mainly among subjects in the Given = Selector group.

Most subjects showed a clear understanding of the two principal dimensions of the $\lambda - N$ space. Their protocols are filled with comments about "using longer programs," "using a different value of N," and so forth. At a finer grain of analysis, subjects were also aware of the importance of what might be called "good instrumentation"—designing programs that have identifiable markers in them. We already saw one such example in subject JS (Appendix B, lines 007–009). The following statements by other subjects are typical (emphasis added):

> I don't want to have two of the same move in there yet, *I might not be able to tell if it was repeating the first one or if it was doing the next part of my sequence* (AD03).
>
> I'm just going to make up some random but different directions *so that I'll know which ones get executed* (RS22).
>
> I'm going to use a series of commands that will ... *that are easily distinguished from one another* and won't run it off the screen (GM27).
>
> ...so I'm going to pick two (commands) that are the direct opposite of each other, to see if ...they don't really have to be direct opposites but ...anyhow, I'm just going to write a program that consists of two steps, *that I could see easily* (BB04).

In addition to working in both the $\lambda - N$ space and the instrumentation space, subjects were generally sensitive to pragmatic constraints, such as using small values of N on commands, so that BigTrak's behavior could be easily observed and remembered.[7]

Although many subjects could articulate these general strategies, they could not always carry them out, as the following selection from subject MA indicates. MA was in the most difficult Selector → Counter condition. He was given D: Repeat last N steps, and the Actual rule was B: Repeat last step N times. MA expressed some doubt about the Given hypothesis and articulated a good experimental strategy:

7. Although it is not shown in this chapter, such awareness of pragmatic constraints contrasts markedly with the behavior of middle-school children in the same situation (Dunbar, Klahr, & Fay, 1989).

> OK, if it repeats the last N steps—which we are presuming, it may or may not do that—if it does, then you'd want to write a program which would have a certain amount of steps before the Repeat key, and not repeat all of them, so that you could see if it actually does that. I'm also going to add steps after it so that, if it repeats the steps after it, you'll be able to see that. So the problem is just making up steps that you can differentiate between, so that's what I'm going to do.

Unfortunately, he decided to write a program that included only FIRE commands, and, although he expressed some doubt about whether he would be able to interpret its behavior, he proceeded as planned:

> I'll have it, OK, well, I was thinking of having it not move anywhere, just fire, but I don't know if I'll be able to tell those apart on the screen. So why don't I do that anyway? So I'm just going to fire once, I'll fire twice, I'll fire 3 times, and then I'll repeat the previous 2 steps, then I'll have it fire four times, then fire 5 times. [FIRE 1, FIRE 2, FIRE 3, REPEAT 2, FIRE 4, FIRE 5].

At this point BigTrak fired 21 times, but MA counted 23 FIREs, got confused, and abandoned the all-FIRE approach to experimentation.[8]

In addition to these general characteristics of individual programs, many subjects were systematic in the *sequence* of programs that they wrote, following, as suggested earlier, a strategy of varying only one thing at a time (such as changing either λ or N but not both from one experiment to the next). We will present more data on this issue in a later section.

4.2 Overall Difficulty

As predicted, subjects were less successful at discovering Selector rules than Counter rules. For the two Given conditions, only 1 of the 18 subjects with Actual-Counter rules failed to discover the rule, whereas 5 of 18 failed to discover Selector rules. The proportion of successful subjects in each condition was: Counter → Counter, 100%; Selector → Counter, 92%, Counter → Selector, 67%, Selector → Selector, 83%. Thus, discovering a Counter rule was easiest, whereas discovering a Se-

8. In fact, all four rules are distinguishable under this program. Rules A, B, C, and D would FIRE 27, 21, 17, and 20 times, respectively. However, this is extremely difficult for the subject to figure out under these circumstances.

lector rule was more difficult. Also, switching from the Selector frame to the Counter frame was much easier than switching from the Counter frame to the Selector frame. Given that we already knew that subjects regard Counter hypotheses as more likely than Selectors (Klahr & Dunbar, 1988), the results of this study suggest that it is the a priori strength of belief in hypotheses that will determine how difficult it is to switch frames.

4.2.1 TRIAL OF THE CORRECT HYPOTHESIS

Another aggregate measure of the relative difficulty of the four conditions is the trial on which subjects arrived at the correct rule, that is, the point when the Current and Actual hypotheses became the same. As shown in the protocol listed in Appendix A, subjects usually stated the Current hypothesis just before they wrote an experiment to test it. Thus, we can compute the proportion of subjects who arrived at the correct hypothesis prior to each experiment. Figure 3 shows the cumulative proportion of subjects in each condition who stated the correct hypotheses prior to the Nth experiment in their series. The effects of condition are very clear. Although a few subjects immediately rejected the Given hypothesis and luckily guessed the Actual rule prior to the first experiment, there is no reliable effect for such correct anticipations. By the second experiment, half of the subjects in both Actual = Counter groups had proposed the Actual rule, but none of the subjects in the Counter → Selector group had.

Recall that subjects were asked to "write three good experiments" to discover how RPT worked. Thus, the proportion of subjects correctly identifying the rule by experiment 3 provides a measure of success on the task as initially presented. As Figure 3 shows, all Counter → Counter subjects could make the minor revision in the preferred hypothesis necessary to go from Rule A to B or B to A by their third experiment. However, when subjects had to change from a Counter to a Selector only 42% of them were able to abandon a preferred Counter for a Selector by experiment 3. As noted above, the difficulty of frame change was asymmetric, as all but one-quarter of the Selector → Counter subjects discovered that the unpreferred Selector was wrong and discovered the correct counter. Finally, even though no frame change was required, subjects had difficulty making the minor within-frame revision necessary in the Selector → Selector condition, and 33% of them failed to do

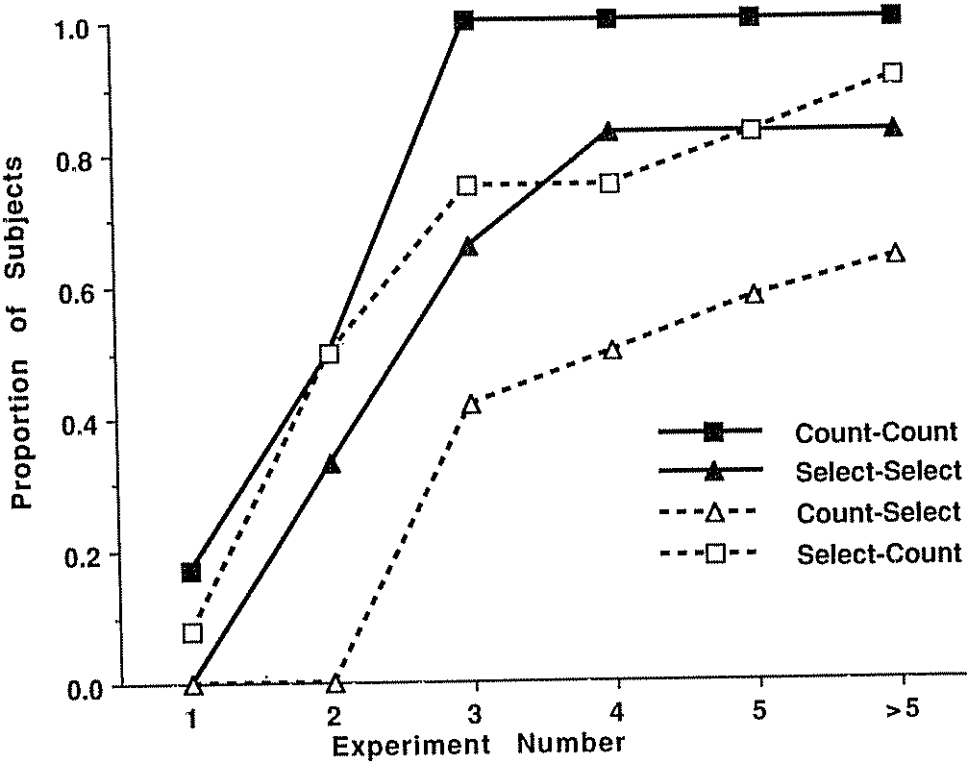

Figure 3. Proportion of subjects generating correct hypothesis by the Nth experiment.

so by the third experiment. This relative order of difficulty remained beyond experiment 3: Counter → Counter was relatively easy, Counter → Selector was relatively difficult, and the two Given = Selector conditions were roughly equivalent and of intermediate difficulty.

4.2.2 THE NUMBER OF EXPERIMENTS

The success-rate measures indicate that the frame change required by the Counter → Selector condition was particularly difficult. Another sensitive measure of difficulty is the number of experiments run. Recall that, once subjects completed the three mandatory experiments, they were free to run additional experiments until they were satisfied that

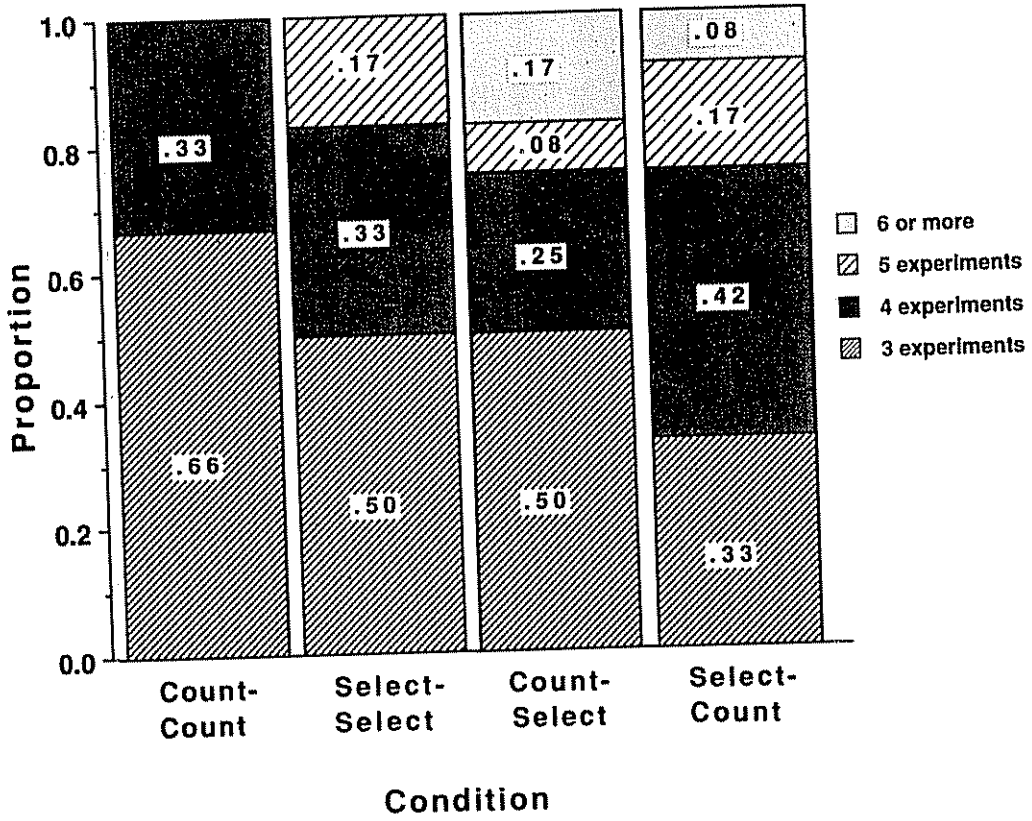

Figure 4. Proportion of subjects running N experiments.

they had discovered the correct rule for RPT. As shown in Figure 4, only one-third of the Counter → Counter subjects chose to run a fourth experiment, and none ran more than four. Half of the subjects in both the Counter → Selector and Selector → Selector conditions, and two-thirds of the Selector → Counter subjects ran four or more experiments. More of the subjects in between-frame conditions ran extra experiments than did subjects in within-frame conditions. The mean number of extra experiments per subject was 0.5 for the within-frame conditions and 1.3 for the between-frame conditions.

4.2.3 IDENTICAL EXPERIMENTS

When subjects were particularly surprised or confused by an experimental outcome, they occasionally repeated an experiment, that is, wrote a program with the same $\lambda - N$ combination as an earlier (usually immediately preceding) program. Although this was an uncommon event, it provided another sensitive index of the relative difficulty of our experimental conditions. Of the 150 total experiments, we observed 14 such pairs of identical experiments, and they occurred only in the frame-change conditions. For both Selector → Counter and Counter → Selector, there were seven repeats. In most of these cases, the problem was that subjects misencoded the outcome of the first experiment, not because it was particularly complex but because their expectations at some crucial point left them unprepared to notice an essential piece of behavior of the device.

4.3 Search in the Experiment Space

Although the legal range of values for both λ and N was from 1 to 15, subjects tended to be conservative in both the length of program they ran and the value of N. Over 90% of the experiments were within the $\lambda \leq 6$ by $N \leq 5$ experiment space depicted in Figure 2, and more than 60% were within a 4 by 3 subset of that range, even though it represents only 5% of the full space. Each experiment was classified according to its location in the $\lambda - N$ space shown in Figure 2. If subjects were selecting values of λ and N at random, then the expected relative frequency of experiments in each of the experiment space regions would be proportional to the size of that region in the 6×5 experiment space (region 1, 6/30; region 2, 10/30; region 3, 14/30) and would be the same for all conditions. If subjects were sensitive to the interaction between the potential informativeness of different regions of the experiment space and the hypothesis being tested, then we would expect to see an effect of frame-type and experiment space region. More specifically, when the goal of hypothesis testing is to demonstrate an effect, subjects should design experiments that will highlight that feature. For Counter hypotheses, this focus would lead to an attempt to demonstrate that N controls the number of repetitions, which is best demonstrated by larger values of N. For small values of λ, this tends to produce programs in region 3. On the other hand, the clearest way to demonstrate a Selector hypothesis is to use a value of N that disambiguates the selected

segment or step from first, last, or all steps in a program. Region 2 is the preferred region for such demonstrations.

4.3.1 EXPERIMENT SPACE DISTRIBUTIONS

Figure 5 shows the distribution of first and third experiments under two different aggregations. The upper panels show the percentage of subjects whose first experiments were in each cell of the experiment space, as a function of the frame of the Given hypothesis. The lower panels show the distributions of the third experiment as a function of whether the subjects were in frame-change or same-frame conditions. For each of the upper panels there were 18 subjects and for the lower panels there were 24 and 11.[9] In the first experiment, region 2 is underrepresented for subjects testing Counter hypotheses and overrepresented for Selector hypotheses, and the opposite is true for region 3. Collapsing over all cells in a region, the two distributions are significantly different from each other ($\chi^2 = 13.6, p < .005$) and from a random model. These results show that, even in their first experiments, the subjects were sensitive both to the properties of the experiment space and to the plausibility (to the subjects) of the Given hypothesis. When given a plausible Counter, subjects focused on number of repetitions rather than on what was to be repeated. This produced programs with large values of N (for Counters, 39% of first experiments had $N > 2$, versus only 17% for Selectors) and relatively short programs (one-third of the Counters but none of the Selectors had $\lambda = 1$ on their first program.) On the other hand, when given an implausible Selector, subjects were likely to start with an experiment that could clearly discriminate between Counters and Selectors. These are longer programs with small values of N (region 2 experiments), and they were used by 55% of the Selector subjects but only by 6% of the Counter subjects.

Recall that, for half of the subjects, the Given hypothesis was from a different frame than the Actual hypothesis. Consequently, classification of experiments by frame of Given hypothesis became increasingly invalid as subjects started to discover that the Given hypothesis was incorrect and designed experiments to test a Current hypothesis from the Actual frame. Therefore, for the third experiment, programs were classified ac-

9. One of the same-frame subjects wrote a program that *began* with RPT, so only two of his experiments are included here.

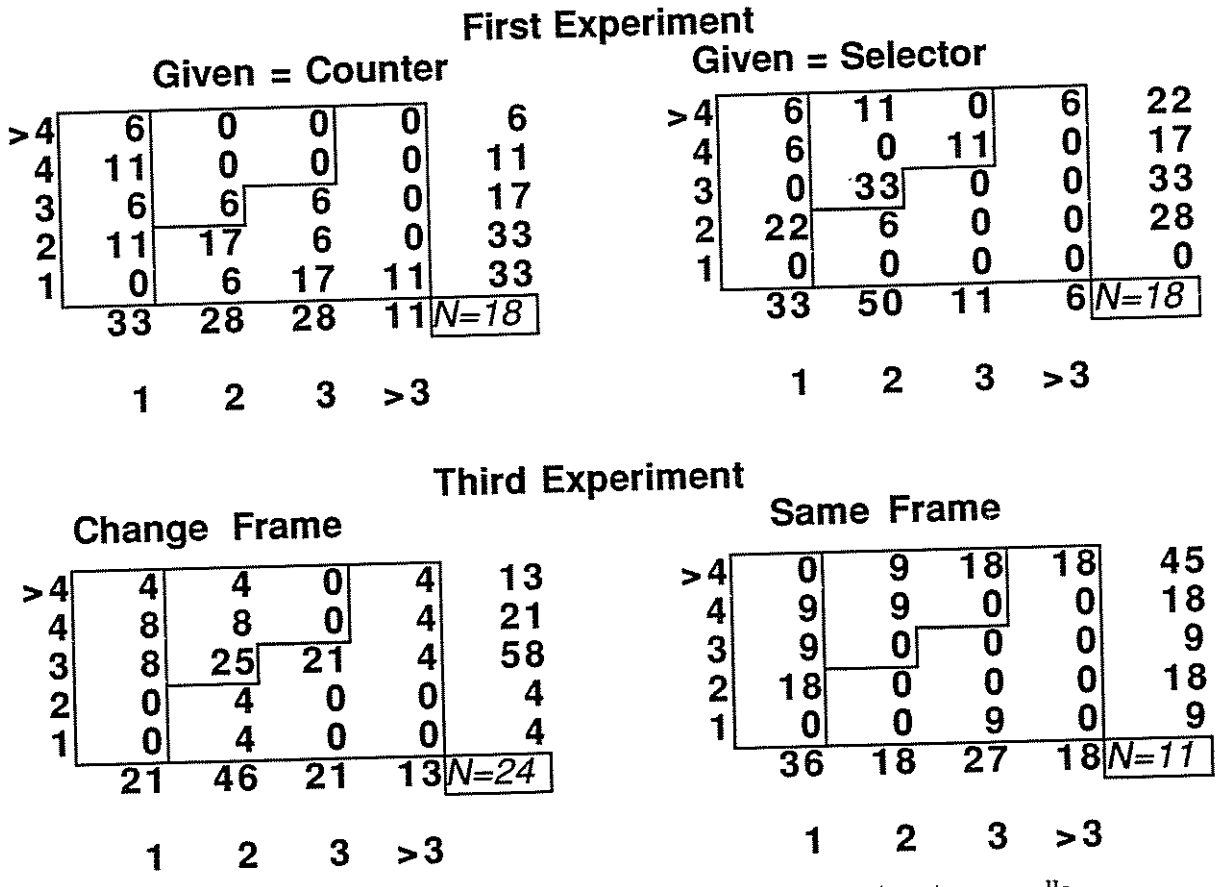

Figure 5. Distribution of first and third experiments in experiment space cells. Rows correspond to λ values; columns correspond to N values. (Entries show percentage of subjects with experiment in that cell.)

cording to whether or not a frame change was necessary. If we aggregate over all the cells in a region, then by the third experiment the frame change by region distributions were not significantly different from one another or from a random distribution of experiments in experiment space regions. However, a cell-by-cell analysis reveals a strong effect of frame change. For the third experiment, 46% of the frame-change subjects, but none of the same-frame subjects, had experiments in cells 3,2 or 3,3. These cells tended to be selected as a consequence of the high discriminability of 3,2 and the incremental VOTAT strategy described in the next section. By experiment 3, subjects in the relatively difficult change-frame conditions were avoiding short programs (only 8% had $\lambda < 3$), and the same-frame subjects show a bimodal distribution for λ. Change-frame subjects are also more consistent in their choice of N (nearly 70% with N equal to 2 or 3).

To determine the effect of experiment space region on overall success rate, we analyzed the data according to whether subjects ever went into region 2 and what region subjects were in just prior to their announcement of the correct hypothesis. There were two kinds of very clear regional effects. First, of the 30 subjects who were successful, 28 went into region 2 at least once (93%), and 4 of the 6 subjects who failed to reach the correct hypothesis never went into region 2 (67%). The two who did go into region 2 wrote programs that did not discriminate between the Actual hypothesis and an idiosyncratic hypothesis that they held. Second, with respect to the region preceding the correct hypothesis, Actual = Counter subjects were in region 2 55% of the time, and Actual = Selector subjects were there 71% of the time. Only 4 of the 30 successful subjects were in region 3 immediately prior to announcing the correct hypotheses. Of these 4, 3 were in Actual = Counter groups where an experiment in region 3 would be sensitive to variations in N, and therefore highly informative.

4.3.2 INCREMENTAL SEARCH IN THE EXPERIMENT SPACE

The analysis of experiment space regions gives a picture of the properties of experiments in isolation, but it does not reflect the nature of the incremental paths followed by subjects as they moved from one experiment to the next. The VOTAT strategy mentioned earlier would lead to conservative moves in the experiment space that do not vary both λ and N at the same time (including moves that vary neither). Overall,

Table 5. Proportion of conservative transitions in experiment space.

CONDITION	FIRST TRANSITION	ALL TRANSITIONS
COUNTER-COUNTER	0.33	0.43
SELECTOR-SELECTOR	0.50	0.44
COUNTER-SELECTOR	0.67	0.64
SELECTOR-COUNTER	0.67	0.64
MEAN	0.54	0.53

about half of the experiment space moves were conservative, but they were more conservative in the frame-change conditions. Table 5 shows the proportion of conservative moves for each condition. The first column shows the proportion only for the first transition (between the first and second experiments), and the second column shows the proportion of all transitions that were conservative. It is clear that, for the Counter → Counter condition, when both the Actual and the Given hypotheses were from the preferred frame, subjects were relatively bold in proposing their second experiment, and two-thirds of them changed both λ and N. However, in frame-change conditions, where the outcome of subjects' first experiment was highly discrepant with their expectations based on the Given hypothesis, subjects were much more conservative in moving about the experiment space; only one-third of them changed λ and N simultaneously.

4.3.3 THE DISCRIMINATING POWER OF EXPERIMENTS

In Section 2.3, we presented a formal analysis of the discriminating power of the different regions of the experiment space. In this section, we summarize the discriminating power of the experiments actually run by the subjects. Each experimental outcome was coded in terms of how many hypotheses were consistent with it. For each subject on each experiment, we considered only the four hypotheses used in this study plus any idiosyncratic hypotheses that the subject might have mentioned. Then we computed, for each condition, the mean number of hypotheses that would be consistent with each experimental outcome (averaged over all the subjects in the condition.) The results are listed in Table 6.

Table 6. Mean number of hypotheses consistent with experimental outcomes.

	EXPERIMENT NUMBER			
CONDITION	1	2	3	4
COUNTER-COUNTER	1.5	1.0	1.2	1.0
SELECTOR-SELECTOR	1.7	1.3	1.2	1.0
COUNTER-SELECTOR	2.8	1.8	1.6	1.5
SELECTOR-COUNTER	1.6	1.3	1.1	1.1

Whereas three of the groups were able to start with programs consistent with only one or two hypotheses, the Counter → Selector subjects designed experiments at the outset whose outcomes were consistent with between two and three hypotheses, and even by their third experiment they were just approaching the first experiment mean of the other three groups.

Another way of describing the discriminating power of subjects' search of the experiment space is in terms of the regions that were avoided while testing particular hypotheses. For first experiments, all subjects avoided an $N = 1$, $\lambda = 1$ experiment as it would not discriminate among any of the hypotheses. Two-thirds of the Given = Counter subjects conducted experiments that could distinguish between the other Counter hypothesis, suggesting that they were testing more than one hypothesis at a time and were avoiding undiscriminating regions of the experiment space. All the Given = Selector subjects conducted first experiments in regions that would discriminate between one selector hypothesis and another.

These results suggest that, when given an hypothesis to test, subjects did consider other hypotheses within that frame, and they wrote programs that would allow them to discriminate between same-frame alternatives. If subjects were only considering hypotheses within the frame of the given hypothesis, then we should expect to see many experiments that would not distinguish between hypotheses from different frames. In fact, 47% of first experiments cannot rule out specific hypotheses from the alternate frame. If we break this down further, we find that, when given Counters, only 33% of first programs can rule out (or confirm) Selectors, whereas, when given Selectors, 66% of programs

could rule out (or confirm) Counters. Again, this reflects the a priori belief that RPT works like a Counter, an important factor in determining what parts of the experiment space to search.

4.4 Response to Discrepancies

One rough measure of the extent to which an hypothesis is incorrect is the difference (Δ) between the number of commands that actually get executed and the number that were expected to be executed under the Current hypothesis. For example, consider program 5 in Table 2. If the Current hypothesis is A and the Actual is B, then $\Delta = 6$. If the Current-Actual pair is B-D, then $\Delta = 0$, and if it is C-D, then $\Delta = 1$. When $\Delta = 0$, there remains the possibility that the *content* of the experimental outcome is discrepant with the prediction, but when Δ is nonzero there is no uncertainty—the prediction is not supported by the outcome. Subjects appear to be sensitive to the size of Δ, even though it abstracts over particular program content. In 69% of the experiments where $\Delta = 0$, subjects changed hypotheses, but in 82% of the cases where $0 < \Delta \leq 2$, and on 100% of the cases where $2 < \Delta$, subjects changed their hypotheses.

5. Discussion

Our subjects were remarkably adept at designing and interpreting experiments in a novel domain. When given a plausible hypothesis, they tended to design experiments that demonstrated the effect that was to be expected. When given implausible hypotheses, they wrote programs that were good discriminators. When the discrepancy between the Given and the Actual hypothesis was very great, subjects were conservative in moving from one experiment to the next. The fundamental question for builders of computational models of the experimental design process is how subjects bring to bear general heuristics for "good experiments" in this novel domain.

5.1 Hypothesis-Generation Heuristics

Any scientific enterprise is conducted in the context of the currently available knowledge of the domain, and initial hypotheses are determined by the knowledge of the domain. In the case of the BigTrak

domain, almost all the commands that were learned in the initial phase work by executing a command N times. As a result, subjects were initially predisposed toward hypotheses that are Counters. This is evident in the results of this study and our previous work (Klahr & Dunbar, 1988). The study discussed in this chapter also suggests that subjects considered more than one hypothesis at a time—both the subject protocols and the types of experiments conducted suggest that the subjects considered various hypotheses within a frame. Thus, one heuristic used is that of generating a frame and then generating various slot values within that frame. Then experiments are conducted that will discriminate between rival hypotheses within the frame. The data also suggest that it is easy to think of hypotheses from an alternate frame, but only when the strength of belief in the current frame is less than that of the alternate frame. Thus, subjects in our Selector → Counter group were much more successful than in the Counter → Selector group.

These findings suggest that a useful heuristic in a computational model of experiment generation would be to initially generate different frames and conduct experiments that distinguish between frames, rather than designing experiments that discriminate between rival hypotheses that are all from the same frame. This heuristic is slightly different from the one that is usually used in discussions of scientific methodology. The usual claim is that multiple hypotheses should be considered when designing an experiment, but here we are arguing that this is most effective when the alternate hypotheses come from different frames. Once the frame is established, then the correct slot values of the frame can be determined. Essentially, we are advocating a form of breadth first search.

One interesting and unexpected result of this study is the fact that subjects tended to test multiple hypotheses, whereas subjects in our previous work and in the work of others (such as Mynatt, Doherty, & Tweney, 1977) generally avoided testing multiple hypotheses. In this study, subjects were given hypotheses to test, whereas in most other studies subjects must generate their own initial hypotheses. This difference in procedure had two effects. First, subjects almost always generated hypotheses other than the one given, resulting in the testing of multiple hypotheses. Second, subjects abandoned the given hypothesis much more readily than if they had generated the hypotheses themselves. In the Klahr and Dunbar (1988) study, most subjects' initial (self-generated) hypothesis was A. They only discovered that RPT

worked according to rule D after 15 experiments. In the present study, two of the three subjects in the A → D group discovered that RPT worked according to rule D after only four experiments. These results suggest that self-generated hypotheses are given higher strength values than externally generated hypotheses—a fact that becomes apparent when articles are submitted for publication!

5.2 Experiment-Generation Heuristics

The BigTrak domain may appear relatively simple in comparison with that faced by a scientist in a laboratory, but the size of the BigTrak experiment space is surprisingly large. Counting only commands and not their numerical arguments as distinct, there are over 30 billion distinct programs (5^{15}) that subjects could choose from for each experiment. Even if we limit the space to programs of length less than or equal to 4, there are nearly 800 different experiments to choose from ($5^4 + 5^3 + 5^2 + 5$). Most subjects appear to understand immediately that specific instructions are not important and that only the $N - \lambda$ space is relevant, but even it can be as large as 225 cells (15×15). Thus, when asked to write only three experiments, subjects must prune this space effectively. There is clear evidence that subjects do manage to drastically prune the space. As noted earlier, 60% of the experiments occurred within a $\lambda \leq 4$, $N \leq 3$ area of the experiment space, although it represents only 5% of the 15×15 experiment space. Even within this preferred area, experiments were not uniformly distributed. The 1, 1 cell was never used, presumably because subjects realize that it provides no information. Conversely, the 3, 2 cell was disproportionately selected 18 times out of 106 total programs in the first three experiments. This is five times more than expected in a random selection from a 6×5 space, and twice the expected frequency in a 4×3 space. This cell represents the minimum values of λ and N in the maximally informative region 2.

What enables subjects to be so effective in constraining their search in the experiment space? We believe that the following heuristics are operating:

1. *Maintain observability.* Given that the BigTrak moves along the screen from one location to another, there is no permanent record of behavior, and subjects must remember what the device actually did. Thus, one heuristic is to write short programs, making it possible

to remember what happened and to compare the results with those predicted by the Current hypotheses. Other uses of this heuristic are to use small values of N to move forward or backward (this is easy to see, and the BigTrak does not go off the screen) and to make turns that are easy to see, such as right-angle, and 180-degree turns.

2. *Design experiments giving "characteristic" results.* In the BigTrak domain, this translates into "use distinct commands." When all the commands in a program are the same, it is extremely difficult to discriminate between rival hypotheses (see the protocol from subject MA quoted earlier). Almost all subjects attempted to write programs where every command was different. This makes it possible to determine what specific commands were repeated as well as the order in which they were repeated. This heuristic substantially reduces the size of the experiment space while maximizing the observability of the programs.
3. *Focus on one dimension of an hypothesis.* Most hypotheses are complex entities and have many aspects that can be focused on. Auxillary hypotheses, ancillary hypotheses, and additional assumptions that are not tested must be made (see Lakatos & Musgrave, 1970). That is, in going from an hypothesis to an experiment, what is thought to be crucial will be focused on. Our results show that in the BigTrak domain subjects tend to focus on one dimension of an hypothesis at a time. For example, when given a Counter hypothesis, subjects initially focused on the number of times something was repeated rather than what is repeated. This heuristic means that subjects miss some of the features of an experimental result as they are considering the result only in terms of the current dimension of the hypothesis that is being focused on. Furthermore, the finding that many experiments changed only one feature of the experiment at a time suggests that the focus was not on only one aspect of an hypothesis but also on one aspect of an experiment. As we mentioned previously, this strategy has been often discussed in the concept attainment literature (Bruner, Goodnow, & Austin, 1956; Tschirgi, 1980).
4. *Exploit surprising results.* Kulkarni and Simon (1988) argue that this heuristic was used by Krebbs in his discovery of the Ornithine cycle, and they have instantiated the heuristic in their program KEKADA. Holland, Holyoak, Nisbett, and Thagard (1986) also note that the generation of new hypotheses from surprising findings (abduction) is

a useful inductive procedure, and they have instantiated it in their PI program. Our results suggest that our subjects also used such a heuristic, but solid evidence for it remains elusive.

One problem is that "surprise" itself is not well defined. If it is defined as *any* discrepancy between expected and observed outcomes, then the heuristic loses much of its power. We have already described one rough measure that could be used to define surprise, Δ, the difference between the expected and actual number of commands executed by a program (see Section 4.4). It is clear that subjects do not ignore Δ, once it is big enough. The identical experiments (Section 4.2.3) provide additional support for subjects' tendency to respond to surprising results. Recall that identical experiments occurred only in the frame-change conditions, which is where subjects are most surprised by the qualitative nature of the discrepancy between the expected and the actual outcome.

Subjects' verbal protocols suggest that they respond to surprise by setting up a new goal of tracking down its source. The successful subjects in the Counter → Selector condition used this heuristic. They focused on why the program or step was not repeated N times and changed their goal from trying to fit the result into a Counter frame to using the surprising experimental result to induce new hypotheses. Subjects in this condition who did not use this strategy continued to focus on how many times things were repeated rather than focusing on the surprising result. Dunbar (1989), in a study that simulated a discovery in a genetics experiment, also found that only subjects who used the strategy of generating a new goal of explaining surprising results were able to discover the mechanism underlying genetic control. This shift of focus usually produces a shift to a new region of the experiment space. The outcome of the next experiment, in turn, leads to the generation of new hypotheses. We have also discussed a group of subjects that use this heuristic (Experimenters) in Klahr and Dunbar (1988).

5. *Use the a priori strength of an hypothesis to choose an experimental strategy.* One of the most often discussed issues in the literature on scientific reasoning has been that subjects tend to attempt to confirm rather than disconfirm their current hypothesis (see Klayman & Ha, 1987). Our study reveals that the strategies of confirmation and disconfirmation varied with the strength of the belief in the currently held hypothesis. When the hypothesis was thought to be

highly likely, subjects often set themselves the goal of demonstrating the key features of the given hypothesis rather than conducting experiments that could discriminate between a large number of hypotheses. A less common strategy for highly likely hypotheses was to use the RPT key as a subgoal to perform an action, for example, drawing a square. In another study (Dunbar, Klahr, & Fay, 1989), we found that young children frequently use this strategy. For hypotheses with low a priori strength, subjects usually propose hypotheses from frames other than the Given frame and conduct experiments that will discriminate between rival hypotheses. Subjects search the hypothesis space before conducting any experiments, and, when they design an experiment, they select an experiment that is in a region of the experiment space that can potentially disconfirm the hypothesis that they are testing.

Not only do subjects appear to use these heuristics, but they also appear to be able to deal with their inherent contradictions. As we noted earlier, no subject ever used the 1, 1 cell, even though it would yield the easiest to observe behavior, because it is so uninformative with respect to discriminating among rival hypotheses. The frequent use of the 3, 2 cell represents a "minimax" solution to the conflicting heuristics of minimizing cognitive load and maximizing discriminability. We are not suggesting that subjects are able to carry out an optimization algorithm that selects this solution. Instead, we believe that the interaction of multiple heuristics produces in our subjects the same kind of behavior that Giere (1988) describes in terms of "the scientist as satisficer."

6. Conclusion

As we stated at the beginning of this chapter, our work falls into the category of psychological studies of scientific discovery in simulated contexts. Our simulated situation is not designed to mimic any particular real scientific discovery, but rather to create a situation in which the thinking processes of subjects are similar to those of scientists when working on real problems. Clearly, a subject's discovery of how RPT works is of scant scientific import. However, we believe that the BigTrak context does give us some insight into the psychology of scientific discovery. In particular, the study described here was designed to contrast high and low plausibility hypotheses (Counters versus Selectors) and minor versus major theory changes (same-frame versus frame-change

conditions). Our results suggest a number of powerful heuristics that can be used to design experiments and formulate new hypotheses. Some of these heuristics are very successful and lead toward discovery. For example, generating hypotheses from alternative frames and setting new goals of explaining surprising results led toward the discovery of the correct hypothesis and resulted in fewer experiments. Other heuristics that tended to be less effective were searching for confirmation and focusing on hypotheses within one frame.

Some of the "good" heuristics that we have discovered are similar to those that have been discovered in other approaches to scientific reasoning that we mentioned earlier—historical analyses of scientific discovery (Darden, 1987) and computational models (Holland, Holyoak, Nisbett, & Thagard, 1986; Kulkarni & Simon, 1988; Langley, Simon, Bradshaw, & Zytkow, 1987). This is encouraging as it suggests that we are coming closer to an understanding of the processes underlying scientific discovery. However, as Klayman and Ha (1987) have noted, certain hypothesis-testing methods that are useful in one context may be totally inappropriate in other contexts. Thus, a further goal for our research is to discover the contexts under which heuristics should and should not be used.

Acknowledgements

This research was supported in part by the Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research, under Contract No. N00014-86K-0349, in part by the A. W. Mellon Foundation, and in part by grant number OGP0037356 from the National Sciences and Engineering Research Council of Canada. We thank Robert Siegler for convincing us that our previous draft was not the final draft.

References

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A study of thinking.* New York: Science Editions, Inc.

Darden, L. (1987). Viewing the history of science as compiled hindsight. *Artificial Intelligence, 8*, 33–41.

Dunbar, K. (1989). Scientific reasoning strategies in a simulated molecular genetics environment. *Proceedings of the Eleventh Annual Meeting of the Cognitive Science Society* (pp. 426–433). Ann Arbor, MI: Lawrence Erlbaum.

Dunbar, K., & Klahr, D. (1989). Developmental differences in scientific discovery strategies. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon.* Hillsdale, NJ: Lawrence Erlbaum.

Dunbar, K., Klahr, D., & Fay, A. L. (1989, April). Developmental differences in scientific reasoning processes. Paper presented at the biennial meeting of the Society for Research in Child Development, Kansas City, MO.

Farris, H., & Revlin, R. (1989). The discovery process: A counterfactual strategy. *Social Studies of Science, 19*, 497–513.

Giere, R. N. (1988). *Explaining science: A cognitive approach.* Chicago, IL: University of Chicago Press.

Gorman, M. E. (1989). Error, falsification and scientific inference: An experimental investigation. *The Quarterly Journal of Experimental Psychology, 41A(2)*, 385–412.

Gorman, M. E., & Carlson, B. (1989). Can experiments be used to study science? *Social Epistemology, 3*, 89–106.

Holland, J., Holyoak, K., Nisbett, R. E., & Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery.* Cambridge, MA: MIT Press.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science, 12*, 1–55.

Klayman, J., & Ha, Y. (1987). Confirmation, disconfirmation and information in hypothesis testing. *Psychological Review, 94*, 211–228.

Klayman, J., & Ha, Y. (1989). Hypothesis testing in rule discovery: Strategy, structure, and content. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*, 596–604.

Kuhn, D., Amsel, E., & O'Loughlin, M. (1988). *The development of scientific thinking skills.* New York: Academic Press.

Kulkarni, D., & Simon, H. A. (1988). The processes of scientific discovery: The strategy of experimentation. *Cognitive Science, 12*, 139–175.

Lakatos, I., & Musgrave, A. (Eds.). (1970). *Criticism and the growth of knowledge.* New York: Cambridge University Press.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). *Scientific discovery: Computational explorations of the creative processes.* Cambridge, MA: MIT Press.

Minsky, M. (1975). A framework for representing knowledge. In P. H. Winston (Ed.), *The psychology of computer vision* (pp. 211–277). New York: McGraw-Hill.

Mynatt, C. R., Doherty, M. E. & Tweney, R. D. (1977). Confirmation bias in a simulated research environment: An experimental study of scientific inference. *Quarterly Journal of Experimental Psychology, 29,* 85–95.

Newell, A. (1989). Putting it all together: Final comments. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon.* Hillsdale, NJ: Lawrence Erlbaum.

Qin, Y., & Simon, H. A. (in press). Laboratory replication of scientific discovery processes. *Cognitive Science.*

Schauble, L. (1990). Belief revision in children: The role of prior knowledge and strategies for generating evidence. *Journal of Experimental Child Psychology, 49,* 31–57.

Shrager, J. (1985). *Instructionless learning: Discovery of the mental model of a complex device.* Doctoral dissertation, Department of Psychology, Carnegie Mellon University, Pittsburgh, PA.

Shrager, J., & Klahr, D. (1986). Instructionless learning about a complex device. *International Journal of Man-Machine Studies, 25,* 153–189.

Simon, H. A., & Lea, G. (1974). Problem solving and rule induction: A unified view. In L. W. Gregg (Ed.), *Knowledge and cognition* (pp. 105–128). Hillsdale, NJ: Lawrence Erlbaum.

Tschirgi, J. E. (1980). Sensible reasoning: A hypothesis about hypotheses. *Child Development, 51,* 1–10.

Tweney, R. D., & Yachanin, S. A. (1985). Can scientists assess conditional inferences? *Social Studies of Science, 15,* 155–173.

Wason, P. C. (1960). On the failure to eliminate hypotheses in a conceptual task. *Quarterly Journal of Experimental Psychology, 12,* 129–140.

APPENDIX A

A Complete Protocol for Subject DP26

CLR and GO commands have been deleted.
BigTrak's Behavior is shown in boldface type.

Given: A — Repeat entire program N times.
Actual: C — Repeat Nth step once.

002 What I want to do is um...
003 I want to test to see if repeat repeats the statement before it,
004 so I'll write just a little program that does something,
005 say turns left ten, and then put the repeat statement in
006 and we'll just repeat it twice 'cause it's not really important
007 for the first program.
008 And then just to check I'll put in a statement after the repeat
009 to see if it has any effect on that,
010 so it's just left 10, repeat it twice and then right ten.
011 EXP: OK, SO WHAT ARE THE COMMANDS?
012 Left ten, repeat two, and right ten.
013 Experiment 1: ← 10 RPT 2 → 10

← 20 → 10

014 It did what you would expect it would do,
015 it did repeat the left ten twice so it turned left twenty
016 and then back to the right ten,
017 so it appears that the repeat doesn't have any effect on any statements
018 that come after it,
019 so I guess now we'll try putting more than 1 statement
020 in front of the repeat and see what happens
021 we'll do left ten and forward two, repeat twice,
022 and just to check it again at the end we'll have it fire once
023 and see if it repeats or not
024 It's left ten, forward two, repeat twice, fire one.
025 Experiment 2: ← 10 ↑ 2 RPT 2 FIRE 1

← 10 ↑ 4FIRE1

026 Ok, it only turned left once,
027 but it went forward four,

028 so it looks like it only repeats the statement immediately in front
029 of it.
030 I guess for the last test we'll just try a more complicated program
031 with the repeat in the middle of it
032 and see what statements are repeated
034 So this is just a long program with a repeat in the middle of it
035 to see what statements are repeated.
036 EXP: OK, SO WHAT ARE THE COMMANDS?
037 Forward one, right fifteen, back one, repeat two, left ten,
038 forward 1.
039 Experiment 3: ↑ 1 → 15 ↓ 1 RPT 2 ← 10 ↑ 1
040

↑ 1 → 15 ↓ 1 → 15 ← 10 ↑ 1

046 Ok, that time it repeated the uh.. the right fifteen command
047 EXP: SO DO YOU KNOW HOW IT WORKS,
048 OR WOULD YOU LIKE TO WRITE MORE PROGRAMS?
049 I'm not really sure how it works,
050 because it seemed to act differently in number two, and number 3.
051 It would be best to write more.
052 Should I just go ahead?
053 EXP: YES JUST WRITE NUMBER FOUR.
054 Let's just try the same program as before,
055 except we'll put a three in, repeat three instead of repeat 2,
056 and see if that has anything to do with it.
057 Forward one, right fifteen, back one, repeat three, left ten,
058 forward one
059 Experiment 4: ↑ 1 → 15 ↓ 1 RPT 3 ← 10 ↑ 1
060

↑ 1 → 15 ↓ 2 ← 10 ↑ 1

061 That time it repeated the statement here but it repeated it twice,
062 the number four.. the number three..
063 it repeated the turning statement once.
064 EXP: WOULD YOU LIKE TO TRY ANOTHER PROGRAM?
065 I guess, I don't really have any idea of what it's doing
066 because when I change the number not only did it change..
067 it didn't change the uh..
068 the number that it repeated
069 but it changed the uh.. the actual instruction.

070 I guess my conjecture is, right now, that it says repeat two
071 so it repeats the second instruction,
072 and here it repeats three and it repeats the third instruction.
073 So we'll try the same thing with repeat one,
074 and see if it repeats the first instruction.
075 Forward one, right fifteen, back one, repeat one, left 10, forward one
076 Experiment 5: ↑1 → 15 ↓1 RPT 1 ← 10 ↑1
077

↑1 → 15 ↓1 ↑1 ← 10 ↑1

078 Ok, I think I know what it does now.
079 EXP: OK..
080 When it hits the repeat statement..
081 when it says repeat one it means at this point repeat statement
082 number one
083 and in this case it went forward and it turned and it went back
084 and then it came forward again, which is the first statement.
085 and it did something similar, I mean it went forward one, turned right
086 went back, and it hit repeat three and this is the third statement
087 so it went back to here
088 EXP: OK, SO HOW, IN GENERAL, DOES THE REPEAT KEY WORK?
089 If you type, it looks, when it hits the repeat statement,
090 if you look through the program when there's like repeat six
091 it takes the sixth statement and does that,
092 when it hits the repeat statement it'll repeat the sixth statement.
093 EXP: OK, GREAT.

APPENDIX B

A Complete Protocol for Subject JS02

CLR and GO commands have been deleted.
BigTrak's Behavior is shown in boldface type.

Given: A — Repeat entire program N times.
Actual: C — Repeat Nth step once.

002 Alright, Program 1, if that is the hypothesis,
003 which I'm not so sure, if it's not the hypothesis
004 I'm going to design a program
005 that's going to prove that it's not the hypothesis,
006 and I think a good way of doing that would be a simple program,
007 so, uh.. I'm going to put in first move forward one
008 and uh.. that's just a good way to start off
009 and I want distinct steps here to see if it is repeating it
010 so I will have a right turn, fifteen degrees
011 then I think a good maneuver here would be just to have it fire once
012 it's just something that's distinguished.
013 Then to see if this thing moves like this
014 it might go in a reverse order
015 or it might just repeat the step number
016 but I sort of doubt that
017 because there's no numbered lines to these programs
018 I'm going to have it move backwards one
019 and that will put it back to the left facing forward
020 and then we will try a repeat which will..
021 we'll try to repeat one
022 repeat one will bring it back to its original position
023 but it will be facing the opposite direction
024 so after the repeat one
025 to see what happens to the instructions that happen afterwards
026 we will put a turn left, thirty degrees
027 EXP: I'M GOING TO HOME IT AND CLEAR IT, NOW YOU TELL ME WHAT TO PRESS
028 Ok, up one, to the right fifteen, fire one, backwards one, repeat one
030 left thirty.
031 Experiment 1: ↑1 → 15 FIRE 1 ↓1 RPT 1 ← 30

032 EXP: NOW WHAT DO YOU THINK MIGHT HAPPEN WHEN I PRESS GO
033 Um, well I think it's definitely going to execute the first part of it
034 it's going to end up facing to the right
035 but over one block to the left of the position it's in now
036 and, uh, then if the hypothesis for the repeat is correct
037 then it will continue with the rest of the program
038 if it's not, the only thing I'm thinking it might do
039 is I think it might just move forward 1
040 and then it'll end up turning to the left 30, reversing it's direction
041 EXP:OK, I'M PRESSING GO

↑1 → 15FIRE1 ↓1 ↑1 ← 30

042 Aha, that's what I thought it would do
043 but that's not what the hypothesis said.
044 EXP: SO WHAT ARE YOU THINKING?
045 Well it's the original idea,
046 it's uh.. if I ran this same program and I said repeat two
047 it would repeat the second step.
048 if I said repeat three, it's going to fire again.
049 It's repeating the order of the steps that I put in,
050 I think.
051 Or, it might,
052 I want to try something here,
053 EXP: WHAT ARE YOU THINKING?
054 Well I'm thinking it might also be..
055 It'll repeat the first step..
056 If I put two, it might repeat the first and the second step
057 so I'm going to try two
058 actually I'll try three.. no I'll try two.
059 Do I have to write this whole program in again?
060 EXP: WHATEVER PROGRAM WE'RE GOING TO DO, YOU NEED TO WRITE IT FOR ME
061 I want to run the same program,
062 because I know what it does,
063 I just want to change the condition of the repeat
064 because I want to see if it's going to repeat
065 the first two instructions
066 or it's just going to repeat the second instruction
067 so we will give it a.. actually we'll give it a three

068 because if it's the first condition
069 then, um.. if it's my first idea
070 it's going to repeat just the third step
071 then I'll have to worry about it turning fifteen degrees
072 it just, it'll be easier
073 EXP: HOME CLEAR NOW WHAT?
074 Up one, to the right fifteen, fire one, backwards one, repeat three,
075 to the left thirty
076 Experiment 2: ↑ 1 → 15 FIRE 1 ↓ 1 RPT 3 ← 30

↑ 1 → 15FIRE1 ↓ 1FIRE1 ← 30

077 So it's just repeating the step number of the..
078 the number you put after the repeat it repeats that sequence,
079 that third (unintelligible) the fire or the turn right
080 or the turn left
081 it doesn't repeat first second and third like I thought it might,
082 it just repeats the third step.
083 EXP: CAN YOU WRITE ONE MORE PROGRAM TO BE SURE?
084 Yeah I'll write one more program.
085 Ok, this has some interesting things,
086 we'll make it move backwards three,
087 we will make it turn to the right sixty,
088 and we'll make it turn to the left..
089 no we want it to go.. we'll have it go straight ahead two
090 and we'll have it fire.. fire five times
091 and then we'll have it, let's see what I want to repeat here..
092 and then we'll have it do a nice little spin,
093 I'm curious, one two three four
094 fifth instruction, we'll make it..
095 (unintelligible) the sixth instruction,
096 I doubt it but I'm curious
097 because that would be the one that's after this,
098 One two three four five, um..
099 and the sixth instruction will be um..
100 what could we make it do interesting..
101 We don't have any backward.. yes we do have a backwards
102 something different, we will make it turn left ten
103 EXP: HOME CLEAR
104 Backwards three, right sixty, forward two, fire five, repeat six,

105 left ten
106 Experiment 3: ↓ 3 → 60 ↑ 2 FIRE 5 RPT 6 ← 10

↓ 3 → 60 ↑ 2FIRE10 ← 10

107 I wonder if it did that because repeat six,
108 since six didn't occur yet,
109 I should have put another step in there,
110 because six didn't occur yet
111 it might not actually be repeating the sixth one,
112 it may just be going on,
113 but that doesn't disprove anything anyway, it's just a thought
114 EXP: SO WHAT ARE YOU THINKING?
115 Well I think that whatever number you put after
116 it repeats that instruction line,
117 I set up this third program to prove that again,
118 but what I was curious about when I designed this program,
119 is whether it would repeat something it actually hadn't done yet.
120 Do you know what I'm saying?
121 Because so far it had moved backwards and turned around
122 and gone forward, it had fired five times,
123 then I'm asking it to repeat the sixth step in the program,
124 but the sixth step hadn't occurred.
125 Now what I should have done,
126 is I should have included another instruction
127 I should have had it repeat the seventh step
128 and put in a sixth instruction that was different,
129 because I don't know, from my last program,
130 I don't know whether,
131 I know that the number six means it'll repeat the sixth instruction
132 but, since it hadn't done it yet,
133 I don't know whether it went to the sixth one..
134 because of the repeat six,
135 or it said repeat six is an illegal quantity to put in there,
136 therefore we go on to the next instruction
137 and it just did the sixth instruction anyway
138 EXP: ARE YOU REALLY SURE YOU KNOW HOW IT WORKS,
139 EXP: OR DO YOU WANT TO WRITE ANY OTHER PROGRAMS TO BE SURE?
140 I'm really sure..
141 I'm curious about the last thing

142 whether it will actually repeat something that hasn't occured yet
143 EXP: BUT YOU'RE FAIRLY SURE YOU KNOW HOW IT WORKS?
144 yes
145 EXP: OK WHY DON'T WE STOP THERE THEN