DUAL SPACE SEARCH DURING SCIENTIFIC REASONING

Technical Report AIP-13

David Klahr and Kevin Dunbar

Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

29 September 1987

This research was supported by the Computer Sciences Division, Office of Naval Research and DARPA under Contract Number N00014-86-K-0678 and in part by a grant from the Spencer Foundation. A preliminary report on the project was presented at the Twenty-seventh Annual Meeting of the Psychonomic Society, November 1986. We are grateful to Sharon Carver, Mike Gorman, Craig Kaplan, Joshua Klayman, Penny Sanderson, Jeff Shrager, Robert Siegler, and Herbert Simon for their helpful comments on earlier drafts, and we thank Sandra Curry and Sally Risinger for their able assistance with various aspects ot the preparation of this manuscript. Additional comments and suggestions made during the review process by Jim Greeno, Clayton Lewis and Don Norman helped clarify and strengthen the paper. Reproduction in whole or in part is permitted for purposes of the United States Government. Approved for public release; distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
Distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AIP - 13

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Carnegie-Mellon University

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Computer Sciences Division
Office of Naval Research (Code 1133)

6c. ADDRESS *(City, State, and ZIP Code)*
Department of Psychology
Pittsburgh, Pennsylvania 15213

7b. ADDRESS *(City, State, and ZIP Code)*
800 N. Quincy Street
Arlington, Virginia 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Same as Monitoring Organization

8b. OFFICE SYMBOL *(If applicable)*

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-86-K-0678

8c. ADDRESS *(City, State, and ZIP Code)*

10 SOURCE OF FUNDING NUMBERS p400005ub201, 7-4-86

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
N/A	N/A	N/A	N/A

11 TITLE *(Include Security Classification)*
Dual Space Search During Scientific Reasoning

12 PERSONAL AUTHOR(S)
D. Klahr and K. Dunbar

13a TYPE OF REPORT
Technical

13b TIME COVERED
FROM 86Sept15 TO 91Sept14

14 DATE OF REPORT *(Year, Month, Day)*
87 September 29

15. PAGE COUNT
64

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Cognitive Psychology, Scientific discover,
Machine Learning

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

SEE REVERSE SIDE

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT □ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION

22a NAME OF RESPONSIBLE INDIVIDUAL
Dr. Alan L. Meyrowitz

22b TELEPHONE *(Include Area Code)*
(202) 696-4302

22c. OFFICE SYMBOL
N00014

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

Unclassified

Abstract

The purpose of the two studies reported here was to develop an integrated model of the scientific reasoning process. Subjects were placed in a simulated scientific discovery context by first teaching them how to use an electronic device and then asking them to discover how a hitherto unencountered function worked. To do this task, subjects had to formulate hypotheses based on their prior knowledge, conduct experiments and evaluate the results of their experiments. In the first study, using 20 adult subjects, we identified two main strategies that subjects used to generate new hypotheses. One strategy was to search memory and the other was to generalize from the results of previous experiments. We described the former group as searching an hypothesis space, and the latter as searching an experiment space. In a second study with 10 adults, we investigated how subjects search the hypothesis space by instructing them to state all the hypotheses that they could think of prior to conducting any experiments. Following this phase, subjects were then allowed to conduct experiments. Subjects who could not think of the correct rule in the hypothesis generation phase discovered the correct rule only by generalizing from the results of experiments in the experimental phase.

Both studies provide support for the view that scientific reasoning can be characterized as search in two problem spaces. By extending Simon and Lea's (1974) Generalized Rule Inducer, we present a general model of Scientific Discovery as Dual Search (SDDS) that shows how search in two problem spaces (an hypothesis space and an experiment space) shapes hypothesis generation, experimental design and the evaluation of hypotheses. The model also shows how these processes interact with each other. Finally, we interpret earlier findings about the psychology of scientific reasoning in terms of the SDDS model.

Table of Contents

1 Two aspects of scientific discovery

The successful scientist, like the successful explorer, must master two related skills: knowing where to look and understanding what is seen. The first skill -- *experimental design* -- involves the design of experimental and observational procedures; the second -- *hypothesis formation* -- involves the formation and evaluation of theory. Historical analyses of scientific discoveries (e.g., Conant, 1964; Mitroff, 1974) suggest that the interaction between experimental design and hypothesis formation is crucial to the success of the real scientific endeavor and that both activities are influenced by the semantics of the discovery context.

However, this interaction can be quite complex; consequently, the implicit research strategy in most psychological studies of scientific reasoning has been to investigate each skill in isolation and in semantically lean contexts. This strategy has yielded many important findings about distinct stages of the scientific reasoning process, but much remains to be learned about how the stages interact and about how the interaction is influenced by prior knowledge. The goal of the work described in this paper is to extend the earlier laboratory studies by investigating scientific reasoning in a context that requires a rich interaction among the processes of hypothesis formation and experiment design. Based upon the analysis of our subjects' behavior in this situation, we propose a framework that integrates the processes involved in scientific reasoning, and then use it as a basis for reinterpretation of some important issues in the area.

1.1 Laboratory studies of scientific reasoning: Two exemplars

In order to provide a background for our proposed extensions, in this Section we summarize two of the best-known laboratory simulations of scientific reasoning. Consider first the series of investigations stimulated by Bruner, Goodnow and Austin's (1956) elegant work on concept learning (e.g., Bourne and Restle, 1959; Hunt, 1962; Shepard, Hovland, and Jenkins, 1961; Whitman and Garner, 1963). The focus of this work is on how subjects select instances from a predefined set in order to evaluate hypotheses and how they form new hypotheses on the basis of feedback about those instances (e.g., Bower & Trabasso, 1964; Levine, 1966; Restle & Greeno, 1970). Instances usually vary in terms of the values of a set of constant attributes, and the rule to be discovered is an arbitrary combination of these values. Bruner, et al. discovered that even in this relatively simple context, subjects use several different strategies for gathering information about hypotheses and they suggested that the different strategies had different levels of "cognitive strain." This interaction between strategies and short-term memory demands was fully articulated by Gregg & Simon (1967), and Greeno & Simon (1984) provide a brief summary of much of the intervening work on concept induction.

As Bruner et al. argued, the concept-learning task is relevant to real science because it involves two essential components of the scientific reasoning process: the logic of experimentation and strategies for discovering regularities. Unfortunately, this relevance played only a minor role during the next 25 years, as most investigators studied the task for its own sake (Bourne and Dominowski, 1972;Medin and Smith, 1984;Neimark and Santa, 1975). The aim of the work presented here is to return to one of the original motivations for the Bruner work -- the laboratory study of scientific reasoning -- and to extend that paradigm along several dimensions so as to bring it even closer to the real nature of scientific reasoning. First, instead of choosing from a set of pre-defined "experiments" (instances), our subjects will have to design experiments of modest complexity. Second, the mapping between experiments and hypotheses will be non-obvious, whereas in the concept learning task, both instances and hypotheses are described in exactly the same language. Third, feedback from experiments will be multivalued rather than just binary. Finally, we will use a context in which prior knowledge and the semantics of the situation play a role in the content, form and plausibility of initial hypotheses and in the criteria for revising hypotheses. (The "simulated universe" tasks used by Mynatt, Doherty & Tweney [1977, 1978] include similar extensions, although their analysis is focused on the logic of confirmation and disconfirmation.)

The second widely-known example is the "2-4-6" rule-discovery task invented by Wason (1960), and used to study scientific reasoning ever since (Gorman and Gorman, 1984;Mahoney and DeMonbruen, 1977;Tukey, 1986;Tweney, Doherty, Worner, Pliske, Mynat, Gross, and Arkkelin, 1980;Wason, 1962;Wetherick, 1962). Subjects are asked to discover a rule (pre-determined by the experimenter) that will classify sets of numerical triads, are told that "2-4-6" is an example of a triad that follows the rule, and are instructed to generate their own triads in attempting to discover the rule. (The experimenter's rule is typically "any increasing series," but subjects usually propose several much more constrained and complicated hypotheses before discovering the correct rule.) The experimenter provides yes/no feedback about instances and also tells subjects whether or not their proposed hypotheses are correct.

The basic finding from these and related studies is that when subjects design experiments, they show a pervasive confirmation bias (Mynatt, Doherty, and Tweney, 1977; 1978).. They propose a single hypothesis and seek evidence that will confirm, rather than disconfirm, it. Mahoney and DeMonbreun (1977) found that scientists and non-scientists did not differ in this regard. The phenomenon is both important and puzzling, and we shall return to it at the end of this paper. A common interpretation of such behavior is that it reveals fundamentally inadequate scientific reasoning skills, but Klayman and Ha (1987) provide a lucid and convincing analysis showing that this characterization is unwarranted in most cases.

Tukey (1986) suggests that "several philosophies of science can readily be applied" to the interpretation of subjects' performance on the 2-4-6 task, as it captures certain aspects of the scientific discovery situation: both instances and hypotheses can have arbitrary complexity, and subjects create their own instances. Nevertheless, as in the concept-learning paradigm, certain extensions would provide a closer analogy to scientific reasoning. One would be to allow subjects to design experiments rather than generating instances. Second, in the 2-4-6 task there is no ambiguity about when to stop: if the subject states the correct rule, no matter what the evidential basis, the task is over. So another positive extension would be to have subjects determine when they have discovered the correct rule, rather than being told by the experimenter. (A recent study by Gorman, Stafford, and Gorman, 1987, has also relaxed this constraint on the Wason task.) Third, as Klayman and Ha (1985) point out, studies using the 2-4-6 task, in all of their variants to date, do not address questions about the content or meaning of initial hypotheses. As noted above in our comments about the concept learning tasks, one of our goals is to establish a research context that enables us to address these important aspects of scientific reasoning.

1.2 Scientific reasoning: Problem solving or concept formation?

There are two principal characterizations of the process of scientific reasoning. One, exemplified by the Bruner and Wason tasks just cited, we call the *concept-formation view*. The argument here is that much of scientific reasoning consists of forming new concepts on the basis of experimental evidence. This view tends to dominate most of the laboratory simulations of the scientific discovery process. The second view, which we call the *problem-solving view*, is exemplified by Simon's (1977) analysis of the discovery process. Under this view, scientific reasoning is characterized as a search process, similar in structure to any problem-solving task, albeit within a complex search space.

The problem-solving view of scientific discovery has its roots in the Gestalt tradition. For example, Wertheimer (1945) implicates search processes in his historical anecdotes about Einstein and Gauss, and Bartlett (1958) is quite explicit in structuring his discussion of the "thinking of the experimental scientist" in terms of search through a set of knowledge states. Simon's contribution to the discovery-as-problem-solving view was to demonstrate how one could go beyond the search metaphor by explaining the discovery process in terms of an explicit theory of human problem solving (Newell, Shaw and Simon, 1958). This basic idea has since been extended substantially by Langley and Simon and their colleagues in their analysis of major scientific discoveries of the last few centuries (Langley, Simon, Bradshaw, & Zytkow, 1987; Kulkarni & Simon, 1987). Their analysis is based on historical records of practicing scientists working for months or years on a problem, while the focus in this paper is on ordinary people in laboratory studies of an hour's duration at most. One purpose of the present study was to devise and execute an experimental study of scientific reasoning within the discovery-as-problem-solving framework.

1.3 Scientific reasoning as dual search: The Generalized Rule Inducer

In the discussion thus far, we have introduced two dichotomies: one dealing with two phases of the discovery process (hypothesis formation and experimental design), and the other with two frameworks for understanding the psychology of these processes (the concept-learning view and the problem-solving view). Our goal in this paper is to replace both dichotomies with an integrated view of the discovery process. In this section we provide an initial overview of our approach, which we will then elaborate in subsequent sections.

At first glance, the concept-formation and problem-solving approaches appear to tackle radically different aspects of the scientific reasoning process; yet as we will argue throughout this paper, both traditions can be organized into a coherent theory of scientific reasoning. The key to this integration comes from Simon and Lea's (1974) insight that both concept learning and problem solving are information-gathering tasks and that both employ guided search processes. Simon and Lea have shown how a single information-processing system -- called the Generalized Rule Inducer (GRI) -- can account for performance in problem-solving tasks and a range of rule-induction tasks, including concept attainment, sequence extrapolation, and grammar induction. The GRI uses the same general methods for both problem-solving tasks and rule-induction tasks. The main difference between problem-solving and rule-induction is in the problem spaces that are used in the task. The rule-induction tasks require search in two problem spaces: a space of rules and a space of instances. Problem-solving search, however, takes place in a single space: a space of rules.

The distinctive feature of rule-induction tasks is that proposed rules are never tested directly, but only by applying them to instances, and then testing whether the application gives the correct result. In rule induction tasks the subject selects (or is shown) an instance and checks to see whether the instance confirms or disconfirms the rule. Instance selection requires search of the instance space, and changing from one rule to another requires search of the rule space. Because rule-induction requires two spaces, the tests operate in a different space from the hypothesis (rule) generator. Simon and Lea's analysis illustrates how information from each space may guide search in the other space. For example, information about previously generated rules may influence the generation of instances, and information about the classification of instances may determine the modification of rules.

The GRI view makes it possible to characterize some further differences between the previous research on concept formation and problem solving. Because the concept-learning research is concerned with rules derived from well-defined instances, the rule space is usually very simple; it consists of all possible combinations of the values and attributes in the instance space. Even when subjects have some control over instance selection, as in the Bruner et al. (1956) work, the full set of permissible instances is predetermined. In problem-solving experiments, the structure of problem space is usually much more complicated. Rather than being merely the concatenation of a set of given features, it consists of a series of knowledge states that the subject can generate by following a wide variety of strategies.

1.4 Extending GRI to the scientific discovery process

Two extensions are required if we are to effect this proposed integration of the concept-learning and problem-solving views of scientific reasoning. First, we need to study subjects' behavior in situations that more closely resemble the scientist's environment than the traditional laboratory tasks that initially motivated the GRI. Second, we need to extend the GRI to accommodate the added complexity of the new situation.

1.4.1 Task elaboration

With respect to the first extension, we devised a task with a more complicated rule space than that used in most concept-formation experiments. Specifically, we studied the behavior of subjects who were attempting to extend their knowledge about a moderately complex device. Adult subjects worked with a programmable, multi-functioned, computer-controlled robot whose basic functions they had mastered previously. (Details will be provided in Section 2.)

Our analysis focuses on their attempts to discover how a new function operates -- that is, to extend their understanding about the device. Experiment construction involves designing an experiment (i.e., a program) and predicting the device's behavior. The analysis phase involves a comparison between an observation of what the device actually did and what the current hypothesis predicted it would do. Incorrect predictions lead to a revised hypothesis and further experimentation. The cycle terminates when subjects believe that they have discovered how to predict and control the behavior of the device. This task allowed us to observe the interplay between the hypothesis-formation and experimental-design phases of the discovery process.

Shrager (1985) showed that when people encounter a novel device, they bring to bear a wide variety of (often inappropriate) prior knowledge in formulating their initial hypotheses about how the device operates. Given the influential role of prior knowledge on initial hypotheses, the literature on analogical problem solving is relevant to our focus on scientific reasoning. Holland, Holyoak, Nisbett, and Thagard (1986) summarize much of this recent work. They suggest that the underlying mechanism for retrieving appropriate prior knowledge involves the summation of activation propagated from the elements of the current problem to related elements in memory. Although there have been a few studies addressing the issues of what triggers or evokes the appropriate prior knowledge, the process is only partially understood (cf. Gick & Holyoak, 1983; Gentner, 1983; Ross, 1984).

1.4.2 Theoretical extension

With respect to the second, theoretical, extension of GRI, we need to augment the GRI processes in two respects. The mapping we propose is between GRI's two spaces (rules and instances) and the corresponding hypothesis space and experiment space involved in the discovery process. Thus, we propose that scientific reasoning can be conceptualized as a search through two problem spaces: an hypothesis space and an experiment space. This means that, first, we need to account for the identification of relevant attributes, for, unlike the conventional studies, our situation does not present the subject with a highly constrained attribute space for hypotheses. Second, we need a more complex treatment of the instance generator, because in our context it consists of an experiment, its predicted outcome, and the observation of the actual outcome. The details of these extensions will be provided in Section 5.

We would expect subjects who are attempting to discover some new function on a partially-understood device to propose the most plausible and "obvious" hypotheses first and to make a sustained effort to prove such initial hypotheses true. Thus it would be useful to have some characterization of subjects' initial knowledge about the device and about potentially relevant general knowledge. The importance of prior knowledge leads us to the final issue in this introduction.

1.5 The inseparability of knowledge and process

Most laboratory studies of scientific reasoning attempt to minimize -- at every stage of the discovery process -- the mutual influence of *strategy* and *knowledge* for the sake of experimental rigor. That is, one class of investigations deals with the *strategies* used in solving "scientific" problems, such as designing experiments (Case, 1974;Siegler and Liebert, 1975), or formulating hypotheses (Kuhn and Phelps, 1982;Wason, 1960), or evaluating evidence (Robinson and Hastie, 1985;Karmiloff-Smith and Inhelder, 1974). The other class deals with people's *knowledge* about the natural world: pendulums, balance scales, falling bodies, etc. (Kaiser, Proffitt, and McCloskey, 1985;McCloskey, 1983;Stavy, Strauss, Orpaz, and Carmi, 1982). But the separation is highly artificial. In any real scientific reasoning context, substantive knowledge and the form of investigative strategy are mutually influential, and the scientist's knowledge about the topic influences the initial hypotheses, the types of experiments conducted, and the way results are analyzed (O'Brien, Costa and Overton, 1986).

In contrast, our goal is to determine how existing knowledge structures determine the initial hypotheses, experiments, and data analysis in a discovery task and to elucidate the process whereby relevant constellations of prior knowledge influence both the formation of hypotheses and the design of experiments. We will focus on how hypotheses are generated by search for appropriate frames and on how experiments are designed either to fill in unspecified variables in those frames or to explore the permissible range of variables. Furthermore, we explore the process whereby experiments lead to the development of more knowledge, which in turn leads to the development of new hypotheses and different strategies of investigation and analysis.

2 Simulating aspects of the discovery process

In this section, we describe the device about which our subjects have to reason, some earlier research using the device, and our procedure for studying scientific reasoning in the laboratory.

2.1 BigTrak

The device we use is a computer-controlled robot tank (called "BigTrak") that is programmed using a LOGO-like language. It is a six-wheeled, battery-powered vehicle, approximately 30 cm long, 20 cm wide and 15 cm high. Interaction takes place via a keypad on the top of the device, which is illustrated in Figure 1. In order to get BigTrak to behave, the user clears the memory with the **CLR** key and then enters a series of up to sixteen instructions, each consisting of a function key (the command) and a 1- or 2-digit number (the argument), terminated by the **GO** key. BigTrak then executes the program by moving around on the floor.

Insert Figure 1 about here

The effect of the argument depends on which command it follows. For forward (↑) and backward (↓) motion, each unit corresponds to approximately one foot. For left (←) and right (→) turns, the unit is a 6° rotation (corresponding to one minute on a clock face. Thus, a 90° turn is 15 "minutes.") The **HOLD** unit is a delay (or pause) of 0.1 sec, and the **FIRE** unit is one auditory event: the firing of the cannon (indicated by appropriate sound and light effects). The other keys shown in Figure 1 are **CLS**, **CK**, and **RPT**. **CLS** Clears the Last Step (i.e., the most recently entered instruction), and **CK**

ChecKs the most recently entered instruction by executing it in isolation. Using **CK** does not affect the contents of memory. We will describe **RPT** later. The **GO, CLR, CLS,** and **CK** commands do not take an argument. To illustrate, one might press the following series of keys:

CLR ↑ 5 ← 7 ↑ 3 → 15 HOLD 50 FIRE 2 ↓ 8 GO

and BigTrak would do the following: move forward five feet, rotate counterclockwise 42 degrees, move forward 3 feet, rotate clockwise 90 degrees, pause for 5 seconds, fire twice, and backup eight feet.

Certain combinations of keystrokes (e.g., a third numerical digit or two motion commands without an intervening numerical argument) are not permitted by the syntax of the programming language. With each syntactically legal key-stroke, BigTrak emits an immediate, confirmatory beep. Syntactically illegal key-strokes elicit no response, and they are not entered into program memory.

2.2 Previous work with BigTrak

In our initial investigations using BigTrak, subjects were given no preliminary instruction; they were simply handed BigTrak and told to "figure out" how it worked. Based on analysis of subjects' protocols in this "instructionless learning" situation, Shrager and Klahr (1986) sketched a framework for characterizing the learning process, and Shrager (1985) constructed a computer simulation model of how initial hypotheses are formed and refined. The model attempts to form initial hypotheses through a process that Shrager calls "view application," in which previously stored knowledge structures are mapped to specific BigTrak elements. For example, if the "calculator view" is activated, then a mapping is made between BigTrak's keys and calculator keys, and the associated knowledge that calculators have memories is used to hypothesize that BigTrak has one also. Shrager's model focused almost entirely on this first phase of the discovery process. Our goal in this study was to establish a procedure that would enable us to track subjects' behavior through the entire cyclical sequence of stages that comprise the discovery process.

3 Study 1: Discovering a new function

In this study[1], we modified the original Shrager and Klahr procedure in two ways. First, we established a common knowledge base about the device for all subjects, *prior* to the discovery phase. Second, we limited the scope of the subject's task to discovering how a single BigTrak function worked. We instructed subjects about how to use all function keys and special keys, except for **RPT**. Subjects learned about the syntax and semantics of the keys and about how to combine commands into a program to accomplish some goal. All subjects were trained to criterion on the keys described earlier and given a fixed set of tasks to accomplish.

Once the training phase was completed, we entered the instructionless phase. Subjects were told that there is a "repeat" key, that it takes a numerical parameter, and that there can be only one **RPT** in a program. Then they were asked to discover how **RPT** works by proposing hypotheses and evaluating them. We suggest that before going further, the reader do the following: formulate an initial hypothesis about how **RPT** work, and then construct a BigTrak program to evaluate the hypothesis. This will provide a subjective impression of the task facing the subject.

3.1 The influence of prior knowledge

One purpose of the instruction phase was to familiarize subjects with the device and the experimental context so that they could function comfortably in the discovery phase. A more important goal of the instruction phase was to establish a realistic but tractable analog to the real scientific context. Scientific work goes on in the context of previously developed theories that provide a background both for the generation of hypotheses and the design of experiments. Analogously, by the time our subjects encounter the **RPT** key, they have various models about BigTrak's functioning as well as general knowledge about what "repeat" means. In the BigTrak context, three categories of prior knowledge may influence subjects' hypotheses about how **RPT** works.

1. **Linguistic knowledge about the meaning of "repeat."** Subjects know that repeating something means doing it again and that various linguistic devices

[1] We use the term "study" here to distinguish our procedures from our subjects' "experiments" with BigTrak.

are needed to determine both *what* is to be repeated and *how many times* the repetition is to occur. There is some ambiguity about whether the number of repeats includes or excludes the initial execution or occurrence (i.e., does "he repeated it twice" mean two or three utterances of the same sentence?).

2. **Programming knowledge about iteration.** BigTrak is a computer-controlled device, and subjects with some programming experience may draw on knowledge about different kinds of iteration constructs from familiar programming languages. Typically, N plays the role of determining the number of repetitions, while the scope is determined by syntactic devices.

3. **Specific knowledge about BigTrak.** Based on what they learn during the training phase, subjects know that there are two types of keys: regular *command* keys that correspond to specific observable actions and take numerical arguments (↑, →, etc), and *special* keys that take no argument (**CLR**, **GO**, etc.). For all command keys, the numerical argument corresponds to the number of repetitions of a unit event (moving a foot, turning a six-degree unit, firing the cannon once, etc.) Although all command keys have an eventual observable consequence, they do not have an immediate action. Two of the special keys (**CK** and **GO**) do have immediate observable consequences. The former executes the most recently entered instruction and the latter executes the entire program. Two other special keys (**CLR** and **CLS**) change an invisible internal state of the device, but cause no immediately observable action.

We predicted that subjects would have difficulty discovering the correct hypothesis without extensive experimentation because the different knowledge sources suggest misleading and conflicting analogies. In most programming conventions, the element(s) to be repeated follow the repeat indicator, the scope is determined by syntax, and the number of repetitions is controlled by a variable. On the other hand, a common linguistic form for repeat implicitly sets the number of repetitions to a single one and the scope to an immediately preceding entity ("could you repeat that?").

Even the specific experience with other BigTrak commands provides contradictory clues to RPT's precise function. One potential conflict concerns the cues for classification of the type of command. RPT is "regular" in that it takes a parameter and does not cause any immediate action, and yet unlike any other regular key, it corresponds to no particular behavior. Another potential conflict concerns the meaning of the parameter. The subject has to determine whether N corresponds to what is to be repeated or to how many times it is to be repeated. Finally, prior knowledge about special keys may leave subjects uncertain about the domain over which the repeat will occur. For the other special keys the domain is either the entire program (CLR and GO) or the single previous step (CK and CLS), but the domain of RPT remains to be discovered.

3.2 Procedure

Twenty adult subjects participated. They were Carnegie Mellon undergraduates participating in the experiment for course credit. All subjects had prior programming experience in at least one language.

The study consisted of three phases. First, subjects were given instruction and practice in how to generate a good verbal protocol. Next, the subjects learned how to use the BigTrak. The experimenter read the manual to the subject and asked the subject to write specified programs that demonstrated how the BigTrak works. At the end of this second phase, the subject was asked to write a moderately complex program. All subjects mastered the device within about 20 minutes.

The third -- and focal -- phase began when the experimenter asked the subject to find out how the repeat key works. Subjects were asked to speak aloud, to say what they were thinking and what keys they were pressing. All subject behavior during this phase, including all key-strokes, was videotaped. At the outset of this phase, subjects had to state their first hypothesis about how RPT worked before using it in any programs. When subjects claimed that they were absolutely certain how the repeat key worked, or when 45 minutes had elapsed, the phase was terminated.

The experimenter interacted with the subject under the following conditions: If the subject was pressing buttons without talking, the experimenter could ask what the subject was thinking. If the subject forgot to bring the BigTrak back to the starting position, the experimenter would ask the subject to do so. Finally, if the subject tested the same incorrect hypotheses with the same type of program more than 3 consecutive times, the experimenter would suggest writing "a different kind of" program.

3.3 Protocol encoding

In this section, we give a brief overview of how we analyzed the protocols. (Details are available upon request.) The video tapes were transcribed into computer text files that contained all verbalizations, timing information, and all key presses. Individual sessions were segmented into episodes by a program that searched for critical delimiters (**CLR**, **CLS**, **CK**, **RPT**, and **GO**) and computed the extent to which BigTrak's behavior would be consistent with a set of alternative hypotheses (to be described in Section 3.4.2). Explicit statements about how the subject thought the **RPT** key might work were coded as hypotheses. Statements of what might happen once the **GO** had been pressed were coded as predictions. Comments about the behavior of the device once the program had been executed were coded as observations.

To illustrate, we will give an example of the encoding of an entire protocol. (The listing, shown in Table 1, is one of our shortest, because it was generated by a subject who very rapidly discovered how **RPT** works.) At the outset, the subject (ML) forms the hypothesis that **RPT N** will repeat the entire program N times (003-004). The prediction associated with the first "experiment" is that BigTrak will go forward 6 units (010-011). The prediction is consistent with the current hypothesis, but BigTrak does not behave as expected: it goes forward only 4 units, and the subject comments on the possibility of a failed prediction (013). This leads him to revise his hypothesis: **RPT N** repeats only the last step (019). At this point, we do not have sufficient information to determine whether ML thinks there will be one or N repetitions of the last step, and his next experiment (021) does not discriminate between the two possibilities. (We call this kind of hypothesis "partially specified," because of the ambiguity. In contrast, the initial hypothesis stated earlier (003-004) is "fully specified.") However, his subsequent comments (024-025) clarify the issue. The experiment at (021) produces results consistent with the hypothesis that there will be N repetitions (BigTrak goes forward 2 units and turns left 60 units), and ML explicitly notes the confirming behavior (022). But the next experiment (026) disconfirms the hypothesis. Although he makes no explicit prediction, we infer from previous statements (023-025) that ML expected BigTrak to go forward 2 and turn left 120. Instead, it executes the entire ↑ 2 ← 30 sequence twice. ML finds this "strange" (028), and he repeats the experiment.

At this point, based on the results of only four distinct experiments, ML begins to formulate and verbalize the correct hypothesis -- that **RPT N** causes BigTrak to execute one repetition of the **N** instructions preceding the **RPT** (030-034) -- and he even correctly articulates the special case where **N** exceeds the program length, in which case the entire program is repeated once (035-037). ML then does a series of experiments where he only varies **N** in order to be sure he is correct (038-046), and then he explores the issue of the *order* of execution of the repeated segment.

In addition to encoding the verbalizations, we classified the experiments according to their length (λ), defined as the number of instructions prior to the **RPT**, and the value of N in **RPT N**. The eleven experiments in ML's protocol are of four general types, defined by the relation between N and λ: $N = 1$ (14, 21, 54); $\lambda < N$ (12, 26, 29); $\lambda = N$ (46, 66); $\lambda > N$, $N \neq 1$ (40, 43, 63). In Section 3.4.3, we will explain why the first three of these types of experiments produce results that are uninformative, at best, and misleading, at worst, while the latter type are highly informative.

Insert Table 1 about here

3.4 Aggregate results

In this section, we present a coarse-grained summary of the data. First (Section 3.4.1), we provide descriptive statistics about the major *categories* of scientific reasoning: hypotheses, experiments, and reaction to experimental outcomes. This analysis will show that although our subjects are generally successful at this task, their behavior diverges widely from any normative model of scientific reasoning. Then we turn to the specific *content* of these categories -- that is, to the particular hypotheses and experiments that are created during the discovery process. In Section 3.4.2, we describe the most commonly proposed hypotheses about how **RPT** works and introduce the *hypothesis space*, a formal characterization of these hypotheses. We conclude the aggregate analysis by shifting our focus from generating and revising hypotheses to designing experiments. We summarize the key dimensions of subjects' experiments (Section 3.4.3) and introduce the notion of the *experiment space*. In Section 3.5, we describe different strategies used by subjects to search both the hypothesis and the experiment space, and, in Section 5, we describe a formal model of the process.

3.4.1 Overall performance

Nineteen of the 20 subjects discovered how the **RPT** key works within the allotted 45 minutes. The mean time to solution (i.e., when the correct rule was finally stated) was 19.8 minutes (s.d. = 9.6 min). In the process of discovering how **RPT** worked, subjects generated, on average, 18.2 programs (s.d. = 11.5).

Of the 364 programs run by the 20 subjects, 304 were *experiments*; that is, they included a **RPT**. Another 51 programs were *control trials*, in which the subject wrote a program without a **RPT**, ran the program, then added **RPT**, and ran the program again. We label the initial program of the pair -- as the one that does not include a **RPT** -- as the control trial. Another 7 programs we label as *calibration trials*: the subject attempted to determine (or remember) what physical unit is associated with **N** for a specific command (e.g., how far is ↑ 1). Only 2 programs that did not contain a **RPT** were unclassifiable.

We define a "common hypothesis" as a fully-specified hypothesis that was proposed by at least two different subjects. Across all subjects, there were 8 distinct common hypotheses. Protocols were encoded in terms of the fully-specified hypotheses listed in Table 2. Subjects did not always 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, and Table 2 shows the proportion of all experiments that were run in Study 1 while an hypothesis was held.[2] (The final two columns in Table 2 will be described in Section 4.2.)

Insert Table 2 about here

Subjects proposed, on average, 4.6 (s.d. = 1.3) different hypotheses (including the correct one). Fifty-five percent of the experiments were conducted under one of the eight common hypotheses. The partially-specified hypotheses, which account for 3% of the experiments, are defined as those in which only some of the attributes of the common hypotheses were stated by the subject. (E.g., "It will repeat it **N** times.") An

[2] As noted earlier, HS1 in Table 2 is the way that BigTrak actually operates.

idiosyncratic hypothesis is defined as one that was generated by only one subject (e.g., "The number calls a little pre-stored program and runs it off."). Such hypotheses are not listed separately in Table 2. For 28% of the experiments, there were no stated hypotheses. For some experiments classified in the "no hypothesis" category, subjects may have actually had an hypothesis, but failed to state it; however, for reasons to be discussed in Section 3.5, we believe that for most of these cases, subjects did, in fact, conduct experiments without an hypothesis in mind. We will also offer an explanation for partial and idiosyncratic hypotheses.

3.4.2 The hypothesis space

There is no limit to either the number of possible attributes or the number of hypotheses that can be formulated from such attributes. Despite this potential for vast variation, the eight common hypotheses -- which account for over half of the experiments -- deal with only four attributes. We can characterize the common hypotheses shown in Table 2 in terms of these key attributes: The role of N, the type of element to be repeated, the boundaries of the repeated element, and the number of repetitions. The resulting *hypothesis space* is shown in Table 3, together with an abstract test program and an indication (in the rightmost column) of how BigTrak would execute the test program, if it operated according to the hypothesis in question.

Insert Table 3 about here

The hypothesis space can also be represented in terms of "frames" (cf. Minsky, 1975). The basic frame for discovering how **RPT** works is depicted at the top of Figure 2. It consists of four slots, corresponding to the four attributes listed above: n-role, unit of repetition, number-of-repetitions, and boundaries-of-segment. A fully-instantiated frame corresponds to a fully-specified hypothesis, several of which are shown in Figure 2. There are two principle subsidary 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 several attributes. 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 number-of-repetitions is *1*.

Insert Figure 2 about here

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 (e.g., N-role:*counter*), also determines what slots remain to be filled (e.g., number-of-repetitions: *N*), and it constrains the focus of experimentation.

Furthermore, frames enable us to represent the differential importance of different attributes, as the "frame type" becomes the most important attribute, and its "slots" become subordinate attributes. This is consistent with Klayman & Ha's (1985) suggestion that "some features of a rule are naturally more 'salient', i.e., more prone to occur to a hypothesis-tester as something to be considered" (p.11). 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, unit and boundary might be chosen so as to produce HC1 (see Figure 2).

Recall that subjects were asked to state their hypothesis about **RPT** *before* actually using it in an experiment. This enabled us to determine what frame is constructed by prior knowledge. In Section 3.1, we discussed the possibility that linguistic knowledge of **RPT**, programming knowledge about iteration, and specific knowledge about BigTrak should conspire to produce inappropriate analogies to **RPT**. This was indeed the case; no subject started off with the correct rule. Seventeen of the 20 subjects started with the N-role:*counter* frame. That is, subjects initially assumed that the role of **N** is to specify the number of repetitions, and their initial hypotheses differed only in whether the repeated unit was the entire program or the single instruction preceding **RPT** (HC1 and HC2). This suggests that subjects drew their initial hypotheses by analogy from the regular command keys, all of which determine the number of repetitions of a unit.

Having proposed their initial hypotheses, subjects then begin to revise them on the basis of experimental evidence. Subsequent hypotheses are systematically related to initial hypotheses. By representing knowledge in terms of frames we can specify the relation among subsequent hypotheses. Of the 55% of all experiments that were conducted with a fully specified hypothesis, nearly two-thirds (.36/.55) were conducted with N-role:*counter*. As shown in Table 2, these experiments dealt with HC1, HC2, and HC3, which assign N the role of counter; another 10% dealt with HN1 and HN2, which assign it no role at all. When subjects were exploring a particular frame, changes in hypotheses usually differed only in the value of a single attribute. (Indicated by connecting arrows in Figure 2). For example, if subjects were using the N-role:*counter* frame, changing the unit of repetition from *program* to *step* would correspond to a change from HC1 to HC2, or changing the bounds from *prior* to *subsequent* would produce HC3 as the hypothesis. When subjects switch from seeing N-role as counter to seeing it as a selector, there is a change in the values of the N-role slot, the unit-of-repetition slot, the number-of-repetitions slot, and the bounds slot. Thus, whenever there is a shift from one frame to the other, at least three slots must change value simultaneously. Fifteen of the subjects make only one frame change, and four of the remaining five make 3 or more frame changes. This suggests that subjects are following very different strategies for searching the hypothesis space. We will return to this issue in Section 3.5.

By abstracting over the *content* of hypotheses, we can analyze the *logic* of confirmation and disconfirmation. If subjects responded according to the classical norms of the scientific method, they would reject disconfirmed hypotheses and retain confirmed ones (cf. Bower & Trabasso, 1964; Popper, 1959). The first row of Table 4 shows the effects of all 220[3] experimental outcomes on subjects' hypothesis-retention behavior. If subjects were perfectly rational, there would be no cases of rejection following confirmation or retention following disconfirmation. Instead, in 25% (21/84) of the instances where the experimental outcome confirms the current hypothesis, subjects change hypotheses, and in over half (76/136) of the disconfirming instances, they retain the disconfirmed hypothesis. In other words, for the average subject, out of 11 experimental outcomes, there are approximately 4 cases in which a disconfirmed hypothesis is retained, and 1 case in which a confirmed hypothesis is abandoned. This

[3] Recall that about 28% of all 304 experiments were performed without a stated hypothesis. They are excluded from this analysis.

is not to say that subjects are entirely insensitive to confirmation/disconfirmation information, for their responses are far from random (X_1^2 = 8.16, $p < .005$). Nevertheless, they show severe departures from the purported canons of good science. These departures have been reported by other investigators, and we will return to this issue in Section 6.

Insert Table 4 about here

3.4.3 The experiment space

Subjects test their hypotheses by writing programs that include **RPT** and observing BigTrak's behavior. The program thus becomes the experiment. But it is not immediately obvious what constitutes a "good" or "informative" experiment. In attempting to construct experiments, subjects are faced with a problem-solving task that parallels their effort to discover the correct hypotheses, except that in this case search is not in a space of hypotheses, but in a space of experiments. Several characterizations of this space are possible: here we describe two extreme forms.

First, consider the space of all "distinct" programs. How large is it? There are six different commands, and programs can have up to fifteen instructions preceding the **RPT**. This yields nearly 500 billion (6^{15}) distinct programs from which subjects can choose, even if we ignore different values of N for each command. Making the more realistic assumption that subjects will tend to limit their experiments to programs having 3 or 4 instructions yields a sharply reduced space of between two hundred and thirteen hundred (6^3 to 6^4) distinct experiments. If we add the additional constraint that (in order to avoid ambiguity) no command should appear more than once in a program, then there are between 120 and 360 distinct experiments that could be run.

A much more tractable experiment space is one that abstracts over the specific content of programs and retains only the values of N and λ, the length of the program preceding the **RPT**. This characterization is based on the observation that other potentially relevant features of the program -- such as the specific commands in the program, their sequence, or the value of their numerical argument -- tend to play only an indirect role in the informativeness of the experiments. That is, the importance of specific instructions is related only to the observability of their independent effects, rather than to **RPT**. For example, for *all* of the common hypotheses, [↑ 2 → 15 **FIRE** 1] is a better test sequence than [↑ 1 ↑ 1 ↑ 1 ↑ 1].

Within the N - λ space, we identify six distinct regions according to the relative value of N and λ and their limiting values. They are depicted in Figure 3, together with illustrative programs. At the bottom of the figure, we indicate which of the common hypotheses would be confirmed by experiments in each region. Here we define the regions and indicate the general consequences of running experiments in each.

Insert Figure 3 about here

- Region I. One-step programs with N = 1 or 2. Although an incrementalist strategy would suggest that this is a good starting place for exploring the experiment space, such experiments are totally undiscriminating: as shown in Figure 3, they produce behavior consistent with all but HC3 in Table 2. Furthermore, the ambiguous distinction between "repeat once" and "repeat twice," mentioned earlier, is exacerbated with a one-step program. Subjects tend not to expect a difference in performance in this case, and BigTrak does not yield one.

- Region II. Multi-step programs with N = 1. Experiments in this region are consistent with hypotheses of the form "it repeats the previous step," such as HC2 and HN2. They rule out hypotheses that the entire program is repeated once (HN1) or N times (HC1).

- Region III. Programs with at least three instructions and a value of N less than λ and greater than 1. As long as no two adjacent instructions are identical, programs in this region are consistent only with HS1 (the correct hypothesis). For example, the program [↑ 2 → 15 FIRE 4 ← 30 RPT 3] is inconsistent with every common hypothesis except HS1.

- Region IV. Here, λ = N. In addition to HS1, these experiments are consistent with hypotheses that RPT causes a repetition of the entire program (HN1), as well as with HS2 (Repeat first N steps once).

- Region V. In this region, N is greater than λ. In this situation, BigTrak effectively sets N equal to λ, so experiments in this region tend to support the hypothesis that N is irrelevant and that HN1 is the correct hypothesis.

- Region VI. Experiments in this region have one-instruction programs with values of N greater than 2. This region is similar to Region V and also serves as the testing ground for hypotheses that N corresponds to the number of repetitions (HC1 - HC3). These hypotheses are disconfirmed in this region, but some subjects perseverate here nevertheless.

Other formulations are possible, but we will use the N - λ space in our analysis. We do not claim that subjects have this elaborated representation of the experiment space. Instead, it enables us to classify experiments according to the kinds of conclusions that they support.

3.5 Strategic variation in scientific discovery: Theorists and Experimenters

There is abundant evidence that leads us to expect strategic variation in problem-solving -- ranging from Bruner at al.'s discovery of different strategies in the concept-learning task, to more recent work on strategic differences in chess, puzzles and physics problems (Chase and Simon, 1973;Klahr and Robinson, 1981;Larkin, McDermott, Simon and Simon, 1980;Simon, 1975), and even to such apparently straightforward domains as single digit addition (Siegler, 1987). It is not surprising then that analysis of our subjects' protocols yielded two distinct experimental strategies.

As noted earlier, subjects started with the wrong general frame. Consequently, their early efforts were devoted to attempting to refine the details of this incorrect frame. The most significant representational change occurred when N-role was switched from counter to selector and a new frame was constructed. Once subjects made this change, they quickly discovered how the **RPT** key works. How did they do this? Subjects were classified as using one of two different strategies according to how they switched from the N-role:*counter* frame to the N-role:*selector* frame. If subjects induced the correct frame from the result of an experiment in region III of the experiment space, they were classified as experiment-space searchers. For convenience, we will refer to them as "Experimenters." These subjects induced the correct frame by searching the experiment space. Thirteen subjects were classified as Experimenters. The remaining 7 subjects discovered the correct frame not by searching the experiment space, but instead by searching the hypothesis space for an appropriate frame. We call the subjects in this group "Theorists." Theorists did not have to conduct an experiment in region III of the experiment space to induce the correct frame. There were other differences among the two groups, but an experiment in region III immediately prior to switching frames was the operational basis for classification.

3.5.1 Theorists: General strategy

The strategy used by the Theorists was to construct an initial frame, N-role:*counter*, and then to conduct experiments that test the values of the frame. When they had gathered enough evidence to reject an hypothesis, Theorists switched to a new value of a slot in the frame. For example, a subject might switch from saying that the prior step is repeated N times to saying that the prior program is repeated N times. When a new hypothesis was proposed, it was always in the same frame, and it usually involved a change in only one attribute.

For Theorists, construction of a new frame was not preceeded by an experiment in region III, nor was it preceeded by a series of experiments where no hypothesis had been stated. Theorists switched frames by searching memory for information that enabled them to construct a new frame, rather than by further experimentation. Knowing that sometimes the previous step and sometimes the previous program was repeated, the Theorists could infer that the unit of repetition was variable and that this ruled out all hypotheses in the N-role:*counter* frame -- these hypotheses all require a fixed unit of repetition. This enabled Theorists to constrain their search for an N-role that permits a variable unit of repetition. As will be shown in Study 2, subjects can construct an N-role:*selector* frame without further experimentation. Following memory search, Theorists constructed the N-role:*selector* frame, and proposed one of the hypotheses within it. They usually selected the correct one, but if they did not, they soon discovered it by changing one attribute of the frame as soon as their initial N-role:*selector* hypothesis was disproved.

3.5.2 Experimenters: General strategy

Subjects in the Experimenter group went through two major phases. During the first phase, they explicitly stated the hypothesis under consideration, and conducted experiments to evaluate it. In contrast, during the second phase, they conducted many experiments without any explicit hypotheses. Experimenters used a variety of initial approaches. Some proposed new hypotheses by abstracting from the result of a prior experiment, and they proposed many hypotheses. These were the subjects, described in Section 3.4.2, who made more than a single frame change; 4 of them made 3 or more such changes. Others stuck doggedly to the same hypotheses, abandoning them only after much experimentation.

The second phase was an exploration of the experiment space. This can be inferred from the number of experiments conducted without explicit statement of an hypothesis: prior to the discovery of how the repeat works, the Experimenters conducted, on average, 6 experiments without statement of an hypothesis. Furthermore, these experiments were usually accompanied by statements about what would happen if N or λ were changed. By pursuing this approach, the Experimenters eventually conducted an experiment in region III of the experiment space. As described earlier, experiments in this region rule out all the commmon hypotheses and are consistent only with HS1. When the subjects conducted an experiment in this region, they noticed that the last N steps were repeated and proposed HS1 -- the correct rule.

3.5.3 Performance differences

While both groups started off with similar strategies -- using hypotheses to guide search in the experiment space -- they diverged in the way they searched for new hypotheses once the initial hypotheses were abandoned: one group searching the hypothesis space for a new hypothesis, and the other exploring the experiment space to see if they could induce some regularities from experimental outcomes. The consequences of these two approaches show up in a few key performance measures, as shown in Table 5. T-tests were conducted on the seven means in Table 5. Following the procedure suggested by Kirk (1968), the over-all level of significance was set at .05; each individual comparison had to be significant at $p < .006$ to be regarded as significant at the over-all $p < .05$ level. As Table 5 shows, the Theorists took less time to discover how the **RPT** key works than the Experimenters; $t(18) = 3.97$ $P < .0009$. The Theorists also conducted half as many experiments as the Experimenters; $t(18) = 3.09$ $p < .006$. There was no significant difference between the two groups in terms of the number of experiments that were conducted under an explicitly stated hypothesis; $t(18) = 1.63$ $p < .12$. However, the Experimenters conducted significantly more experiments in which an hypothesis was not explicitly stated; $t(18) = 3.70$ $p < .002$. There were no differences between the two groups in the number of different hypotheses stated; $t(18) = 1.83$ $p < .08$, nor in the number of hypothesis switches; $t(18) = 1.93$ $p < .07$. Thus, the major difference between the two groups is in the number of experiments conducted without an explicitly stated hypothesis, which, in turn, accounts for the greater number of experiments conducted by the Experimenters, as well as the length of time to reach a solution.

Insert Table 5 about here

The two groups can be compared in terms of sensitivity to experimental outcomes and the strategies used to explore the experiment space. Consider first the sensitivity to experimental outcomes. In Section 3.4.2, we discussed the aggregate result showing that although subjects departed substantially from normative models, they did show a significant sensitivity to experimental outcomes. The second two rows of Table 4 show the same general pattern for Experimenters and Theorists, with no difference between the two groups (χ^2_3 = 1.49), indicating that neither group responds more (or less) "rationally" to experimental outcomes.

The strategies that the two groups use to search the experiment space also can be compared; subjects can change either N, λ, both N and λ, or neither N nor λ. An analysis of the data in Table 6 also shows that there was no significant difference in how the two groups search the experiment space: χ^2_3 =5.35 p < .20. Although the two groups differ in the conditions under which they decide to search the experiment space, they do not differ in *how* they search it, at this aggregate level of analysis. However, a finer grain analysis reveals an interesting difference between the two groups: The number of different N - λ combinations that are used is a measure of the extent to which the experiment space is explored. There are 225 different N - λ combinations that could be explored but only a small fraction of this experiment space is actually used. Recall that the Experimenters conducted twice as many experiments as the Theorists. If they are using those extra experiments to explore the experiment space more widely than the Theorists, then they should explore more N - λ combinations. This was indeed the case; overall, the Theorists explored 19 distinct N - λ combinations, whereas the Experimenters explored 50. At an individual level, the Experimenters explored significantly more N - λ combinations than the Theorists; *t*(18) = 3.29 p < .004.

Insert Table 6 about here

In summary, the main difference between the Theorists and the Experimenters is that the latter group conduct more experiments than the Theorists and that this extra experimentation is conducted without an explicit hypothesis statement. We have argued that this extra experimentation is indicative of a search of the experiment space, and we have shown that the Experimenters do indeed use more N - λ combinations than the theorists. Furthermore, we have argued that instead of conducting a search of the experiment space, the Theorists search the hypothesis space for an appropriate role for N. This is an important claim for which there was no direct evidence in the protocols. Therefore, we conducted a second study to test the hypothesis that it is possible to think of an N-role:*selector* hypothesis without exploration of the experiment space.

4 Study 2: Hypothesis-space search and experimentation

Our interpretation of subjects' behavior in Study 1 generated two related hypotheses: A: It is possible to think of the correct rule via pure hypothesis-space search, without using any experimental results; B: When hypothesis-space search fails, subjects switch to experiment-space search. In Study 2, we directly investigated each of these hypotheses.

- If Hypothesis A is correct, then it should be possible for subjects to propose the correct rule without the benefit of any experimental outcomes. Study 1 provided no direct evidence for this hypothesis, because no subject in Study 1 mentioned the correct rule without doing at least *some* experimentation. In Study 2. we tested this hypothesis by asking subjects to state not just one, but *several*, different ways that **RPT** might work, *before* doing any experiments. If subjects can think of the correct rule without any experimentation, then this will provide support for the view that the Theorists in Study 1 did indeed construct the appropriate frame without using experimental input. This was the hypothesis-space search phase of Study 2. This phase was followed by the experimental phase, in which the subjects were allowed to conduct experiments as in Study 1. We expected that subjects who mentioned the correct rule during the hypothesis-space search phase would discover the correct rule with relatively little experimentation.

- Hypothesis B asserts that if hypothesis-space search is unsuccessful, then subjects switch to a search of the experiment space. We argued that this was the strategy used by the Experimenters in Study 1. This hypothesis predicts that subjects who fail to discover the correct rule during the first phase of Study 2 should not be able to discover the correct rule by hypothesis-space search during the second, experimental, phase of the task. Thus, we predict that subjects who are unable to generate the correct rule in the hypothesis-space search phase will behave like the Experimenters of Study 1 and will discover the correct rule only after conducting an experiment in region III of the experiment space.

- A further goal of Study 2 was to discover whether generation of several hypotheses prior to experimentation would change the way subjects generated and evaluated experiments. In Study 1, subjects always tested hypotheses one at a time; they never conducted experiments that would distinguish between a number of hypotheses. In Study 2, having considered a number of different hypotheses before entering the experimental phase, subjects may test multiple hypotheses in a single experiment. Generation of hypotheses before the experimental phase may also make the subjects more willing to abandon their preferred hypotheses in the face of inconsistent evidence. If so, then even those subjects who do not generate the correct hypothesis during phase 1 should conduct significantly fewer experiments than the subjects in Study 1. When an hypothesis is disconfirmed they will switch to another (previously generated) hypothesis rather than continuing with the same hypothesis.

4.1 Method

Subjects. Ten Carnegie Mellon undergraduates participated in the experiment for course credit. Five subjects had taken at least one programming course, and the other five had no programming experience.

Procedure. The familiarization part of Study 2 was the same as described for Study 1: subjects learned how to use all the keys except the **RPT** key. Familiarization was followed by two phases: hypothesis-space search and experimentation.

The hypothesis-space search phase began when the subjects were asked to think of various ways that the **RPT** key might work. In an attempt to get a wide range of possible hypotheses from the subjects, we used three probes in the same fixed order:

1. "How do you think the **RPT** key might work?"

2. "We've done this experiment with many people, and they've proposed a wide variety of hypotheses for how it might work. What do you think they may have proposed?"

3. "When BigTrak was being designed, the designers thought of many different ways it could be made to work. What ways do you think they may have considered?"

After each question, the subject responded with as many hypotheses as could be generated. Then the next probe was used.

Once the subjects had generated all the hypotheses that they could think of, the experimental phase began: The subjects were allowed to conduct experiments while attempting to discover how the **RPT** key works. This phase was nearly identical to the discovery phase of Experiment 1, with a few variations in how the data were collected. Instead of videotape recording, we used an audio tape for subjects' verbalizations. Keypresses were also captured on the audiotape by having subjects tell the experimenter what keys to press. Otherwise, the procedure was the same as that used in Study 1.

4.2 Results

4.2.1 Phase 1: Hypothesis-space search

Subjects proposed, on average, 4.2 different hypotheses. All but two subjects began with the N-role:*counter* frame, and 7 of the 10 subjects switched to the N-role:*selector* frame during Phase 1. The correct rule (HS1) was proposed by 5 of the 10 subjects.

The last column in Table 2 shows the number of subjects who proposed each hypothesis at least once. These numbers are only roughly comparable to the other entries in the table (from Studies 1 and 2) because the first two columns indicate the proportion of experiments run under each hypothesis, while the final column is simply frequency of mention (because subjects ran no experiments during the hypothesis-space search phase). Nevertheless, some similar patterns emerge. First, all but one of the common hypotheses of Study 1 was mentioned by at least 2 of the subjects.

Furthermore, as in Study 1, hypotheses HC1 and HC2 were among the most frequently mentioned hypotheses (indeed, all but one subject proposed HC2). However, half of the subjects proposed hypotheses from the N-role:*selector* frame, whereas in Study 1, fewer than 10% of the experiments dealt with hypotheses from the N-role:*selector* frame. It is possible that in Study 1 the information gathered from the exploration of the experiment space may have inhibited subjects from switching to the N-role:*selector*.

4.2.2 Phase 2: Experimentation

All subjects were able to figure out how the **RPT** key works. As can be seen from Table 7 mean time to solution was 6.2 minutes, and subjects generated, on average, 5.7 experiments and proposed 2.4 different hypotheses.

Subjects were again classified as Experimenters or Theorists according to whether or not they discovered the correct rule after conducting an experiment in region III of the experiment space. In this study, there were six Experimenters and four Theorists. The performance of the two groups on a number of measures is shown in Table 7. Note that all of the Theorists stated the correct rule during the hypothesis-search phase and that they all had prior programming experience. In contrast, only one of the Experimenters stated the correct rule during the hypothesis-space search phase, and only one of them had any prior programming experience. While the differences between the means of the two groups on all the measures mirror the earlier pattern from Study 1, none of them are significant, due to the small sample size and large within-group variances.

Insert Table 7 about here

The experiment-space search patterns in this study are radically different from those in Study 1. The Study 2 Experimenters conducted far fewer experiments than either the Experimenters or the Theorists of Study 1. Subjects in Study 2 switch hypotheses more readily; in Study 1, both the Experimenters and the Theorists changed their hypothesis after disconfirmation only 44% of the time. In Study 2 (see Table 8), the Theorists changed hypotheses after disconfirmation 85% of the time and the Experimenters changed after 58%. Furthermore, the proportion of experiments that are conducted in region III of the experiment space is far higher than in Study 1. This indicates that the subjects switched to conducting experiments in region III earlier than the subjects in Study 1.

Insert Table 8 about here

4.3 Discussion

The results of the hypothesis-space search phase of Study 2 show that it is possible for subjects to generate the correct hypothesis (among others) without conducting any experiments. This result is consistent with the view that the Theorists in Study 1 think of the correct rule by a search of the hypothesis space. The results of the experimental phase of Study 2 further support our interpretation of Study 1. All of the subjects who failed to generate the correct rule in the hypothesis-space search phase behaved like Experimenters in the experimental phase: They discovered the correct rule only after exploring region III of the experiment space. This is consistent with the view that when hypothesis-space search fails, subjects must turn to a search of the experiment space.

The differences between the results of Study 1 and Study 2 are striking. The main difference is that subjects conducted far fewer experiments in Study 2. A prior search of the hypothesis space allows the subjects to generate the N-role:*selector* frame much more readily than in Study 1. This is true even for subjects who could not think of the correct rule in the hypothesis-space search phase. Furthermore, subjects in this study did attempt to conduct experiments that allow them to distinguish between two hypotheses. For example, subjects might be trying to distinguish between two hypotheses in the N-role:*counter* frame: "repeats the previous step N times" and "repeats the previous program N times." They will write a program and vary the value of N, this will quickly bring them into region III of the experiment space, and they discover how the **RPT** key works. Subjects in Study 1 rarely designed hypothesis-discriminating experiments, for they usually were dealing with only a single hypothesis at a time. Thus it took them longer to abandon hypotheses, and they conducted few experiments in region III.

The substantial influence of prior knowledge is further demonstrated by the finding that all of the Theorists, but only one of the Experimenters, had prior programming experience. Knowing something about programming allowed the Theorists to construct the correct frame, although precisely what aspect of programming knowledge was crucial here is undetermined. Nevertheless, the interesting finding in this study is that the effect of differential prior knowledge propagates through the initial hypothesis-fomulation stage to influence differences in experimental strategies.

In sum, prior exploration of the hypothesis space had two main effects on the experimental phase. First, it allowed subjects to generate hypotheses that are in the N-role:*selector* frame. As a result, subjects quickly switched to the N-role:*selector* frame in the experimental phase. Second, because subjects were aware that a number of hypotheses could account for their results (even if they were working within the N-role:*counter* frame), they conducted discriminatory experiments. Often the best way of distinguishing between hypotheses is to conduct an experiment in region III of the experiment space. Once subjects conducted such an experiment, they quickly discovered the correct rule.

5 A Dual-Search Model of Scientific Discovery

Recall that the point of departure for our analysis of scientific reasoning is Simon & Lea's Generalized Rule Inducer. GRI was designed to account for the results of traditional laboratory studies of problem solving and rule induction. As noted in Section 1.3, two extensions are necessary in order to apply the concept of dual-space search underlying GRI to the broader and more complex domain of scientific discovery. The first extension involves an enrichment of the complexity, depth, and inter-connectedness of the phases of the discovery task presented to subjects. This extension was described in Section 2 and the results were described in Sections 3.4 and 3.5. We argued that qualitative differences in subjects' behavior could be interpreted in terms of differences in how they allocated their search effort between a space of experiments and a space of hypotheses. The second extension to GRI is a further specification of the processes involved in searching these two spaces. In this section we describe a model that incorporates such extensions.

5.1 SDDS: General description

We start by summarizing the key features of our model of scientific discovery as dual search (SDDS). It is proposed as a general model of scientific reasoning that can be applied to any context in which hypotheses are proposed and data is collected. The fundamental assumption is that scientific reasoning requires search in two related problem spaces: the hypothesis space, consisting of the hypotheses generated during the discovery process, and the experiment space, consisting of all possible experiments that could be conducted. Search in the hypothesis space is guided both by prior knowledge and by experimental results. Search in the experiment space may be guided by the current hypothesis, and it may be used to generate information to formulate hypotheses.

SDDS consists of a set of basic components that guide search within and between the two problem spaces. Initial hypotheses are constructed by a series of operations that result in the instantiation of a frame with default values. Subsequent hypotheses within that frame are generated by changes in values of particular slots, and changes to new frames are achieved either by a search of memory or by generalizing from experimental outcomes. Our description of SDDS will proceed as follows: In Section 5.2, we first introduce the basic components and their evoking conditions. Then in Section 5.3.2, we show how the model accounts for the different strategies described in Sections 3.5 and 4.

5.2 SDDS components

Because we are proposing SDDS as a general framework within which to interpret behavior from any scientific reasoning task, we introduce it at a very general level, without reference to our specific BigTrak context. In Sections 5.3.2 and 6 we will return to an interpretation of our results. Three main components control the entire process from the initial formulation of hypotheses, through their experimental evaluation, to the decision that there is sufficient evidence to accept an hypotheses. The three components, shown at the top of the hierarchy in Figure 4, are SEARCH HYPOTHESIS SPACE, TEST HYPOTHESIS, and EVALUATE EVIDENCE.

- The output from SEARCH HYPOTHESIS SPACE is a fully specified hypothesis, which provides the input to TEST HYPOTHESIS.

- TEST HYPOTHESIS generates an experiment appropriate to the current hypothesis (E-SPACE MOVE), makes a prediction, and observes the outcome. The output of TEST HYPOTHESIS is a description of evidence for or against the current hypothesis. based on the match between the prediction derived from the current hypothesis and the actual experimental result.

- EVALUATE EVIDENCE decides whether the cumulative evidence -- as well as other considerations -- warrants acceptance, rejection, or continued consideration of the current hypothesis.

These processes and their subcomponents are hierarchically depicted in Figure 4, which is described in the following paragraphs.

Insert Figure 4 about here

5.2.1 Search hypothesis: Sub-components

SEARCH HYPOTHESIS SPACE has two components. If there is no active frame, then the system generates one. Usually a new frame has unfilled slots, so the next step is to assign specific values to those slots. If there is an active frame, it may require changes in some slot values.

- GENERATE FRAME has two components corresponding to the two ways that a frame may be generated.

    - o EVOKE FRAME is a search of memory for information that could be used to construct a frame. This is the process in which the wide variety of prior knowledge sources -- discussed earlier -- would influence the formation of hypotheses. We will not attempt a detailed elaboration of how specific knowledge elements are activated on the basis of the current context, for that would occupy an entire volume. The main purpose of isolating EVOKE FRAME in SDDS is to distinguish it from the other possible source of new frames: INDUCE FRAME.

    - o INDUCE FRAME generates a new frame by induction from a series of outcomes.

        - The first sub-process in INDUCE FRAME generates an outcome, and the second process generalizes over the results of that (and other) outcomes to produce a frame. GENERATE OUTCOMES will be described below. The specific termination rule and the mechanism for cumulating outcomes are unspecified. The result from GENERATE OUTCOME is a behavior pattern that is input to

GENERALIZE OUTCOMES, which then attempts to generalize over the outcomes in order to produce a frame.

The distinction between EVOKE FRAME and INDUCE FRAME corresponds to the difference between situations in which subjects are able to recall similar situations and use them as the basis for constructing initial frames, and situations in which subjects must observe some behavior before they can venture even an initial hypothesis.

- The purpose of ASSIGN SLOT VALUES is to take a partially instantiated frame and assign specific values to the slots so that a fully specified hypothesis can be generated. It has two components for which we have not specified a preferred order. Values may be assigned by using prior knowledge (USE PRIOR KNOWLEDGE) or by using specific experimental outcomes (USE EXPERIMENTAL OUTCOMES).

  - If there are already some experimental outcomes, then they can be examined to determine specific slot values (USE OLD OUTCOMES).

  - Alternatively, the system can use GENERATE OUTCOME to produce some behavior solely for the purpose of determining slot values.

  In the early phases of the discovery process, USE PRIOR KNOWLEDGE plays the major role in assigning values, whereas later in the course of experimentation, USE EXPERIMENTAL OUTCOMES is more likely to generate specific slot values. If the system is unable to assign slot values to the current frame (because they have all been tried and rejected), then the Frame is abandoned, and the system returns to GENERATE FRAME.

The end result of SEARCH HYPOTHESIS SPACE is a fully specified hypothesis, which is then input to TEST HYPOTHESIS. Note that "experiments" may be run in two different subcontexts in the service of SEARCH HYPOTHESIS SPACE, and that neither of these contexts involve the evaluation of an hypothesis, for it is still being formed.

5.2.2 Test hypothesis: Sub-components

TEST HYPOTHESIS uses three sub-components to: formulate an experiment (E-SPACE MOVE), make a prediction, and run the experiment.

- E-SPACE MOVE produces an experiment. It will be described below, as it is used in several places in the model.

- MAKE PREDICTION takes the current hypothesis and the current experiment and predicts specific results, centered on the current focal values.

- RUN the experiment, OBSERVE the result, and MATCH to expectation. RUN produces a description of a discrepancy between the prediction and the actual behavior. As depicted here, the expected behavior is generated prior to the running of the experiment (during MAKE PREDICTION). However, SDDS allows the computation of what "should have happened" to occur *following* the running of the experiment, during the MATCH process. MATCH requires descriptions of both the expectation and the observation as input.

TEST HYPOTHESIS outputs a representation of evidence for or against the current hypothesis; this representation is then used as input by EVALUATE EVIDENCE.

5.2.3 Evaluate Evidence

EVALUATE EVIDENCE determines whether or not the cumulative evidence about the experiments run under the current hypothesis is sufficient to reject or accept it. It is possible that the evidence is inconclusive and neither situation obtains, in which case EVALUATE EVIDENCE loops back to TEST HYPOTHESIS. Note that the input to the review process consists of a cumulation of output from earlier TEST HYPOTHESIS cycles. The scope of this cumulation could range from the most recent result, to the most salient ones, to a full record of all the results thus far. The content of this record could be one of either consistency or inconsistency.

Additional factors may play a role in EVALUATE EVIDENCE. For example, *plausibility* seems to distinguish some of adults' and children's hypotheses, particularly those that perform some arbitrary arithmetic operation on N. *Functionality* arguments appear in some of the protocols, and cause subjects to reject hypotheses that give no role to N, even if they have been confirmed (e.g., "why would it take a number if it's not used?", or "why would they design a RPT key in the first place?"). Although these factors appear to influence behavior, we do not yet have a full understanding of how they work.

5.2.4 Generate Outcome

This process It conists of an E-SPACE MOVE, which produces an experiment, RUNing the experiment and OBSERVING the result.

5.2.5 E-space move

Experiments are designed by E-SPACE MOVE. The most important step is to FOCUS on some aspect of the current situation that the experiment is intended to illuminate. "Current situation" is not just a circumlocution for "current hypothesis", because there may be situations in with there is no current hypothesis, but in which E-SPACE MOVE must function nevertheless. (This is an important feature of the model, and it will be elaborated in Section 5.3.2). If there is an hypothesis, then FOCUS determines that some aspect of it is the primary reason for the experiment. If there is a frame with open slot values, then FOCUS will select an one of those slots as the most important thing to be resolved If there is neither a frame nor an hypothesis -- that is, if E-SPACE MOVE is being called by INDUCE FRAME, then FOCUS makes an arbitrary decision about what aspect of the current situation to focus on.

Once the focal value has been determined, CHOOSE sets a value in the Experiment Space that will provide information relevant to it, and SET determines the values of the remaining, but less important, values necessary to produce a complete experiment.

5.3 Comments on the model

5.3.1 Memory requirements

A variety of memory requirements are implicit in our description of SDDS, and must, by implication, play an important role in the discovery process. Here we provide a brief indication of the kinds of information about experiments, outcomes, hypotheses and discrepancies that SDDS must store and retrieve.

- Recall that GENERATE OUTCOME operates in two contexts. Under INDUCE FRAME it is called when there is no active hypothesis, and the system is attempting to produce a set of behaviors that can then be analyzed by GENERALIZE OUTCOMES in order to produce a frame. Therefore, SDDS must be able to represent and store one or more experimental outcomes each time it executes INDUCE FRAME.

- Another type of memory demand comes from EVALUATE EVIDENCE: in order to be able to weigh the cumulative evidence about the current hypothesis, REVIEW OUTCOMES must have access to the results produced by MATCH in TEST HYPOTHESIS. This would include selected features of experiments, hypotheses, predictions, and outcomes.

- Similar information is accessed whenever ASSIGN SLOT VALUES calls on USE PRIOR KNOWLEDGE or USE OLD OUTCOMES to fill in unassigned slots in a frame.

At this point in the model's development, the precise role of memory remains an area for future research.

5.3.2 The multiple roles of experimentation in SDDS

Examination of the relationship among all these processes and subprocesses, depicted in Figure 4, reveals both the conventional and unconventional characteristics of the model. At the top level, the discovery process is characterized as a simple repeating cycle of generating hypotheses, testing hypotheses, and reviewing the outcomes of the test. Below that level, however, we can begin to see the complex interaction among the subprocesses. Of particular importance is the way in which E-SPACE MOVE occurs in three different places in the hierarchy:

1. as a subprocess deep with GENERATE FRAME, where the goal is to generate a behavior pattern over which a frame can be induced,

2. as a subprocess of ASSIGN SLOT VALUES where the purpose of the "experiment" is simply to resolve the unassigned slots in the current frame,

3. as a component of TEST HYPOTHESIS, where the experiment is designed to play its "conventional role" of generating an instance (usually positive) of the current hypothesis.

Note that the implication of the first two uses of E-SPACE MOVE is that in the absence of hypotheses, experiments are generated atheoretically, by moving around in the experiment space.

SDDS also elaborates the details of what can happen during the EVALUATE EVIDENCE process. Recall that three general outcomes are possible: the current hypothesis can be accepted, it can be rejected, or it can be considered further.

- In the first case, the discovery process simply stops, and asserts that the current hypothesis is the true state of nature.

- In the second case -- rejection -- the system returns to H-SPACE SEARCH, where two things can happen. If the entire *frame* has been rejected by EVALUATE EVIDENCE, then the model must attempt to generate a new frame. If EVOKE FRAME is unable to generate an alternate frame, then the system will wind up in INDUCE FRAME and will ultimately start to run experiments (in GENERATE OUTCOME) in order to find some element of behavior from which to do the induction. Having induced a new frame, or having returned from EVALUATE EVIDENCE with a frame needing new slot values (i.e., a rejection of the hypothesis but not the frame), SDDS executes ASSIGN SLOT VALUES. Here too, if prior knowledge is inadequate to make slot assignments, the system may wind up making moves in the experiment space in order to make the assignments. (i.e., GENERATE OUTCOME under USE EXPERIMENTAL OUTCOMES). In both of these cases, the behavior we would observe would be the running of "experiments" without fully-specified hypotheses. This is precisely what we see in the second phase of the Experimenters' behavior (see Section 3.5), and for most of the children.

- In the third case, SDDS returns to TEST HYPOTHESIS in order to further consider the current hypothesis. The experiments run in this context correspond to the conventional view of the role of experimentation. During MOVE IN E-SPACE, FOCUS selects particular aspects of the current hypothesis and designs an experiment to generate information about it.

5.3.3 Extending the model

As yet, SDDS is not a running computer model, rather it is a specification of the control structure for a yet to be built program. The actual building of the model will involve a much more extensive and precise specification of the processes involved. Here we will sketch some of the possible extensions to SDDS based on several of the related ideas that have emerged in the field of Machine Learning.

We need to specify how prior knowledge is activated, searched and utilized by the discovery context. SDDS lumps all these processes under EVOKE FRAME, yet there are a large number of complex processes that are involved in this mapping that we have not addressed. Carbonell's work on derivational analogy (Carbonell, 1986) suggests a number of possible heuristics that could be used in the EVOKE FRAME process. Holland, Holyoak, Nisbett, and Thagard (1986) have also proposed several mechanisms that effect the mapping from prior knowledge to the current experimental context.

Our notion of partially specified hypotheses is similar to the different levels of specificity in Mitchell's (1979) *version spaces.* However, it is not clear whether complex contexts, such as the one we have been studying, will prove as susceptible to the "single representation trick" (Cohen and Feigenbaum, 1982) in which both instances and rules can be expressed in the same representation. As Cohen and Feigenbaum point out, if the trick is inapplicable, then "searches of the two spaces must be coordinated by complex interpretation and experiment planning procedures." (p. 368)

Notable among the models that do not use a single representation for rules and instances are the "BACON series" of programs (Langley, Simon, Bradshaw, and Zytkow, 1987). When provided with the appropriate sets of training instances (which represent the knowledge available to scientists working on the problem at that point in history) BACON and its successors have been able to rediscover several important scientific concepts.

Deciding which experiment to conduct next is obviously an important process. We have only sketched it at a broad level in our description of E-SPACE MOVE and it needs further elaboration. Three approaches to experiment generation that may be relevant to our implementation of SDDS are exemplified by AM, KEKADA and LEX. Lenat's (1977) AM also performs something analogous to experiment planning when it is attempting to collect examples of a concept under refinement, and it uses dozens of general heuristics to search its experiment space. Of particular relevance to SDDS is the way that AM's search is connected to an extensive prior knowledge base. As noted at the outset, we believe that substantive knowledge influences the search in both spaces during the discovery process, and the studies reported here have indicated some aspects of this influence. Kulkarni and Simon (1987) have suggested a number of additional heuristics that scientists might actually use to conduct further experiments. Their Experimentproposer heurisitics are largely domain-specific and we expect that some of our heuristics will be also. However, we believe that there are a number of domain-independent heuristics that can be used, such as those in LEX (Mitchell, Utgoff and Banerji, 1983). LEX has a problem generator that will allow it to formulate the correct hypothesis by generating experiments that discriminate between very general and very specific hypotheses which have been formulated on the basis of previous experimental results (i.e., refining the version space). This type of generator would be a component of MOVE IN E-SPACE.

6 General discussion

Our concern is with both the logic of experimentation *and* the link between the formation of hypotheses and experimental results, so we used a task having a number of distinctive characteristics: (a) The concept to be discovered was moderately complex and was not the concatenation of a number of simple features. (b) Instead of just selecting an instance, subjects had to create experiments that would produce some behavior. (c) Experimental results were the actual behaviors of the device, and the subjects could extract much more than one bit of information from them. (d) The correctness of an hypothesis was never announced, but had to be determined by the subjects' own evaluation of the accumulated evidence. (e) Prior knowledge could influence the strength of initial hypotheses, as well as the ease with which alternatives were generated.

The results of both studies support the view that when subjects attempt to discover how a device works, they must search in two problem spaces: an hypothesis space and an experiment space. In the previous section, we developed a model (SDDS) that embodies this idea. Thus one important feature of SDDS is the way in which it integrates these two searches. A second important feature is the articulation of the multiple roles played by experimentation. In this final section, we will suggest how SDDS can provide a useful framework for understanding scientific reasoning in general. Dual-space search can be used to understand the development of hypotheses (Section 6.1), the logic of experimentation (Section 6.2), and strategy differences in scientific discovery (Section 6.3).

6.1 Hypothesis formation and scientific discovery

One of the central features of SDDS is that it accounts for two different aspects of hypothesis generation: how hypotheses are generated and why on some occasions there are large differences between adjacent hypotheses, while on others there are only minor differences. Consider first how hypotheses are generated. In SDDS, hypotheses can be generated either from prior knowledge or by generalizing from the results of prior experiments. These two possible knowledge sources play a role both in SEARCH HYPOTHESIS SPACE (EVOKE FRAME and INDUCE FRAME) and in ASSIGN SLOT VALUES (USE PRIOR KNOWLEDGE and USE EXPERIMENTAL OUTCOMES). EVOKE FRAME has its strongest effect at the beginning of the task; subjects formulate their initial hypotheses on the basis of the frame(s) most activated by the features of their current focus. Once subjects have exhausted all the relevant values of a frame, they will again use SEARCH HYPOTHESIS SPACE. Some subjects construct a new frame by using EVOKE FRAME, and others construct it by using the results from INDUCE FRAME.

Differences in the degree of similarity between adjacent hypotheses is a consequence of the use of frames. Initial experimentation is directed at the resolution of particular slot values within a frame. The slot values are changed as a result of prior knowledge (USE PRIOR KNOWLEDGE) or experimentation (USE EXPERIMENTAL OUTCOMES) experimentation. This leads to the postulation of hypotheses that differ in only minor respects, as subjects change the values only a few at at time. Thus, when subjects search within a frame there will be only minor differences between adjacent hypotheses. When new frames are generated, there will be large differences between hypotheses: recall that when there is a change of frames there is a change in the types of attributes in all the slots, resulting in a radically different knowledge state.

Previous research can also be interpreted in this manner; for example, Mynatt et al. (1977, 1978) used a task in which subjects had to discover the laws of repulsion and attraction in an arbitrary "physics world." Subjects had to propose hypotheses and generate experiments (firing particles at test objects) to test their hypotheses. While Mynatt et al. were concerned mainly with whether subjects attempted to falsify their hypotheses, their results suggest that their subjects were exploring frames and switching frames after they had exhausted all possible values of the frame. In fact, Mynatt et al. (1978) note that many hypotheses were minor variations on previous hypothesis -- indicating investigation of a frame -- and that there were also occassional large differences in adjacent hypotheses -- indicating a switch to a new frame.

Representational change accompanying a new frame can be viewed as a form of illumination or insight (cf. Wallas, 1926; Duncker, 1945). As Simon (1977) notes, although research on insight commonly assumes that "asking the right question is the crucial creative act," it is more likely that "reformulation of questions -- more generally, modification of representations -- is one of the problem-solving processes" and that "new representations, like new problems, do not spring from the brow of Zeus, but emerge by gradual -- and very slow -- stages."

Our results are consistent with this view. None of our subjects started with the correct general frame. However, once they were driven to it by earlier failed hypotheses and observation of results, they were able to form the correct hypothesis. In other words, results of failed experiments forced subjects to consider the role of N, and this caused a restructuring of the hypothesis space. If restructuring is conceived as generation of a new frame then the nature of insight becomes obvious. Insight is not merely the change of values in slots of a pre-existing frame, rather it is the instantiation of a new frame -- this is what is meant by a restructuring of the representation. The interaction between the experiment space and the hypothesis space plays a crucial role in such restructuring.

6.2 The logic of scientific inference and SDDS

Almost all prior research on scientific reasoning has been concerned with the logic used in reasoning tasks. Researchers have devoted an enormous amount of effort to understanding two aspects of disconfirmation. First, why do subjects fail to test potentially disconfirming instances when evaluating hypotheses? Second, why do subjects fail to change their hypothesis in the face of disconfirming outcomes? A third question, raised by our results, is why subjects change hypotheses that have just been confirmed. In this section, we will suggest how these issues can be interpreted using the SDDS model.

6.2.1 Failure to seek disconfirmation

One of the most robust findings in the scientific reasoning literature is that subjects exhibit a pervasive "confirmation bias." That is, they perfer to select instances that they expect to confirm rather than disconfirm their hypothesis. Klayman and Ha (1987) argue that most people follow a heuristic they call a "positive test strategy" -- "a tendency to examine cases hypothesized to be targets" -- and they show that, when the probability of an hypothesis being confirmed is small, this strategy can provide useful information. Furthermore, a positive test strategy provides at least a sufficiency test of one's current hypothesis. However, if, as in Wason's "2-4-6" task, the probability of confirming one's hypothesis is high, then positive tests will not provide any useful information. Klayman and Ha's analysis suggests that the appropriateness of the strategy depends on the distribution of positive and negative instances, although such information is not available to the subject.

In our task, subjects' almost invariably followed the positive test strategy: They stated that "if BigTrak does this, then my hypothesis is correct." According to Klayman and Ha's argument, our subjects' strategy was appropriate, because for about 60% of the experiments in Study 1 and Study 2, subjects received disconfirming evidence (see Tables 4 and 8). Subjects learned that their initial hypotheses were false and so changed to other hypotheses. Thus even though subjects were looking for evidence that would confirm their hypotheses, the hypotheses were usually falsified.

SDDS provides an interesting extension of this view of confirmation bias. As Klayman and Ha note, subjects' strategies should depend on what they think the nature of the instances they encounter will be; if there are many positive instances a negative test strategy will be more useful than a positive test strategy. Following a positive test strategy and producing a predominance of disconfirming evidence forces subjects to either search memory in order to construct a new frame or search the experiment space for a data pattern that can suggest a new hypotheses. Because the subjects in the Experimenter group discovered that regions II and VI of the experiment space disconfirmed their initial hypotheses, they switched to regions III, IV, and V. A positive test strategy enabled them to avoid further search of uninformative regions of the experiment space.

More generally, a positive test strategy may help scientists in two ways. First, it may enable them to avoid perseveration on incorrect frames by abandoning EVOKE FRAME altogether in favor of INDUCE FRAME. Second, it may influence them to conduct different types of experiments for whatever hypotheses they do hold. Kulkarni and Simon (1987) have argued that Krebs's discovery of urea was prompted by an exploration of the experiment space. Thus, a positive test strategy may be a useful heuristic in the early stages of investigation, as it allows the subject to determine those types of instances that are worthy of further experimentation. In our study, search of the experiment space is initially guided by a positive test strategy, but because so few regions of the experiment space are consistent with initial hypotheses, this strategy provides useful information as to which parts of the experiment space to search next. The generality of this finding remains to be demonstrated, but it suggests some interesting further studies.

6.2.2 Tolerating disconfirming evidence

Recall that our subjects frequently maintained their current hypotheses in the face of negative information. In Study 1, fewer than half of the disconfirming outcomes lead to immediate hypothesis changes. SDDS suggests some possible explanations for this behavior. One contributing factor is the probabalistic nature of the basic processes underlying TEST HYPOTHESIS and EVALUATE EVIDENCE. An unanticipated consequence of the complexity of our procedure was that -- due to the fallibility of memory and of the OBSERVE & MATCH processes -- the feedback subjects received had some probability of error. That is, from the subjects' perspective, there might be error in either the device behavior, their encoding of that behavior, or their recall of the current program and associated prediction. Gorman (1986) demonstrated that when subjects are told that there is some probability of error in the feedback received during a rule discovery task they tend to "immunize" their hypotheses against disconfirmation by classifying disconfirming instances as the erroneous trials. Thus, some cases of perseveration may result from subjects simply not believing the outcome and attributing the apparent disconfirmation to one of several fallible processes. The most likely candidates for this explanation are the cases in which the subject not only retains a disconfirmed hypothesis, but actually repeats the exact same experiment (see 26-29 in Table 1). Another error-related cause of perseveration may be even simpler: subjects erroneously encode the disconfirming behavior as confirming behavior.

The non-deterministic nature of experimental evidence can also have an effect on the decision mechanism in EVALUATE EVIDENCE. This process is based not only on whether the result of the prior experiment rules out the hypothesis, but also on whether enough evidence has accumulated to accept or reject the hypothesis. The amount of evidence in favor of an hypothesis and the strength of the hypothesis both determine when subjects will continue to hold or will switch an hypothesis. Only when the cumulative disconfirming evidence exceeds a criterion will an hypothesis be changed. In the present study, subjects had general sources of prior knowledge that predisposed them to the n-role:*counter* frame. These hypotheses had a high apriori strength and needed much disconfirming evidence to be rejected. However, once the initial hypotheses were rejected, subjects conducted few experiments on subsequent hypotheses. Because these subsequent hypotheses had lower strength, any evidence that appeared to contradict them quickly led to their rejection. Other authors have made similar observations. O'Brien et al., for example, note that "subjects are less likely to take evidence as conclusive when their presuppositions about the content domain discourage them from doing so" (p. 509).

An alternative explanation that has been offered for the finding that subjects tend to stick to disconfirmed hypotheses is that they cannot think of alternative hypotheses. Einhorn and Hogarth (1986), suggest that:

> ... because the goal of causal inference is to find some explanation for the observed effects, the discounting of an explanation by specific alternatives still leaves one with the question, 'If X did not cause Y, what did?' ... In fact, the distinction between testing hypotheses and searching for better ones can be likened to the difference between a 'disconfirmation' versus 'replacement' mode of inference. The replacement view is consistent with the Kuhnian notion that theories in science are not discarded, despite evidence to the contrary, if they are not replaced by better alternatives (Kuhn, 1962). Indeed, the replacement view is equally strong in everyday inference. (pp. 14-15)

The results from our studies provide a basis for elaborating this view. We know that when subjects *do* have alternatives readily available -- as in Study 2 -- they are more likely to drop disconfirmed hypotheses than when they don't -- as in Study 1. On the other hand, when subjects could no longer think of any new hypotheses, they could decide to search the experiment space and not hold any hypotheses at all. Thus, subjects did not have to stick with their hypotheses once they had accumulated enough evidence to reject them, because it was permissible in our study to "replace something with nothing."

6.2.3 Abandoning verified hypotheses

The other side of perseveration in the face of disconfirmation is changing hypotheses in the face of confirmation. Recall that, on average, subjects in Study 1 had one instance in which they changed an hypotheses even though the most recent experimental outcome confirmed it. Strictly speaking, this is not a departure from from logical norms, as positive experimental results can only provide what Klayman and Ha (1987) call "ambiguous verification", rather than "confirmation" as we have been calling it. Our interpretation of this behavior also invokes memory processes, but this time in a positive way. That is, subjects do have memory for previous outcomes, and the current result may not only confirm the current hypothesis, but, when added to the pool of previous results, may be consistent with some other hypothesis that appears more plausible or interesting. In order to fully account for this, SDDS would have to elaborate EVALUATE EVIDENCE so that it could look for global, as well as local, consistency in deciding what to do with the current hypothesis.

6.3 Dual search: The source of different strategies

Strategic differences that have been observed in the prior work on concept formation bear certain similarities to the strategies that we have observed in our studies. Bruner, et al. observed two basic strategies. The first is called focussing: subjects focus on a positive instance and change the values of instances one attribute at a time until they discover the concept's defining values. In terms of our model, Bruner's focussers, by systematically attending to the relevance of individual attributes, were cautiously searching the experiment space. Our Experimenters pursued a strategy similar to focussing. They searched the experiment space, not with an hypothesis in mind, but only with an encoding of the last few instances of the device's behavior. Their goal was to discover the attributes common to all the instances that they generated.

The second strategy that Bruner et al discovered was successive scanning; subjects test a single hypothesis at a time. Both our Experimenters and our Theorists used this strategy, though the difference between our two groups was that the experimenters switched from testing hypotheses to collecting information that would allow them to generate a new hypothesis. Bruner et al. argued that subjects adopt one strategy rather than another because some strategies impose more of a cognitive strain, or Short-term memory load, than others. However in our task the source of difference between Experimenters and Theorists is in Long-term memory: Subjects who can construct the correct frame from information in long-term memory are Theorists. Those who are unable to construct the correct frame from information in long-term memory are Experimenters, and must search the experiment space.

Our Experimenter/Theorist distinction is roughly analogous to the data-driven vs model-driven distinction in AI approaches to inductive inference. However, for most Machine Learning models, both the amount and the accuracy of information required far excedes the capacity of our subjects. This fallibility may account for the kind of strategy differences -- similar to ours -- that have been observed in other discovery tasks. For example, Rasmussen (1981) found two different types of strategies used by operators trying to find faults in a complex system. Some operators search the experiment space trying to find the faulty component. Other operators search a hypothesis space in order to think of a set of symptoms which are similar to the observed symptoms. Rasmussen also found that use of these strategies vary with the amount of knowledge about the domain that the operators have: Experts tend to search the hypothesis space, and novices tend to search the experiment space. It is likely that in Rasmussen's case, as in ours, the different strategies result from differences in prior knowledge rather than from a stable individual difference.

6.4 Conclusion

We have proposed that scientific reasoning requires search in two problem spaces and that the different strategies that we observed are caused by different patterns of search in these two problem spaces. We proposed SDDS as both a framework for interpreting these results and as a general model of scientific reasoning. Clearly, there are many aspects of the scientific reasoning process that we need to specify further, but we believe that SDDS offers a potentially fruitful framework for discovering more about discovery.

7 References

Bartlett, F.C. (1958). *Thinking*. New York: Basic Books.

Bourne, L.E., & Dominowski, R.L. (1972). Thinking. *Annual Review of Psychology, 23*, 105-130.

Bourne, L.E., Jr., & Restle, F. (1959). Mathematical theory of concept identification. *Psychological Review, 66*, 278-296.

Bower, G.H., & Trabasso, T.R. (1964). Concept identification. In R.C. Atkinson (Ed.), *Studies in mathematical psychology*. Stanford, CA: Stanford University Press.

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

Carbonell, J.G. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.), *Machine Learning: An artificial intelligence approach, Volume II*. Los Altos, CA: Morgan Kaufmann Publishers, Inc.

Case, R. (1974). Structures and strictures: Some functional limitations on the course of cognitive growth. *Cognitive Psychology, 6*, 544-573.

Chase, W.G. & Simon, H.A. (1973). The mind's eye in chess. In *Visual information processing*. New York: Academic Press.

Cohen, P.R., & Feigenbaum, E.A. [Eds.]. (1982). *Handbook of Artificial Intelligence, Vol. 3*. Reading, MA: Addison-Wesley Co., Inc.

Conant, J.B. (1964). *Two modes of thought: My encounters with science and education*. New York: Simon & Schuster.

Duncker, K. (1945). On problem solving. *Psychological Monographs, 58*(5), . Whole No. 270.

Einhorn, H.J., & Hogarth, R.M. (1986). Judging probable cause. *Psycholgical Bulletin, 99*, 3-19.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7*, 155-170.

Gick, M. L. & Holyoak, K. J. (1983). Schema induction and analogic transfer. *Cognitive Psychology, 15*, 1-38.

Gorman, M.E. (1986). How the possibility of error affects falsification on a task that models scientific problem solving. *British Journal of Psychology, 77*, 85-96".

Gorman, M.E., & Gorman, M.E. (1984). A comparison of disconfirmatory, confirmatory and control strategies on Wason's 2-4-6 task. *Quarterly Journal of Experimental Psychology, 36A*, 629-648.

Gorman, M.E., Stafford, A., & Gorman, M.E. (1987). Disconfirmation and dual hypotheses on a more difficult version of Wason's 2-4-6 task. *The Quarterly Journal of Experimental Psychology, 39A*, 1-28".

Greeno, J.G., & Simon, H.A. (1984). Problem solving and reasoning. In R.C. Atkinson, R. Herrnstein, G. Lindzey, & R.D. Luce (Eds.), *Stevens' handbook of experimental psychology*. New York: Wiley & Sons.

Gregg, L. W., & Simon, H. A. (1967). Process models and stochastic theories of simple concept formation. *Journal of Mathematical Psychology, 4*, 246-276.

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

Hunt, E.B. (1962). *Concept Learning: An information processing problem.* New York: Wiley & Sons.

Kaiser, M.K., Proffitt, D.R., & McCloskey, M. (1985). The development of beliefs about falling objects. *Perception & Psychophysics, 38*(6), 533-539.

Karmiloff-Smith, A., & Inhelder, B. (1974). If you want to get ahead, get a theory. *Cognition, 3*, 195-212.

Kirk, R.E. (1968). *Experimental design: Procedures for the behavioral sciences.* Belmont, CA: Brooks/Cole.

Klahr, D., & Robinson, M. (1981). Formal assessment of problem solving and planning processes in preschool children. *Cognitive Psychology, 13*, 113-148.

Klayman, J., & Ha, Y. (1985). Hypothesis testing in rule discovery: Strategy and structure. Paper presented at the Tenth Research Conference on Subjective Probability, Utility, and Decision Making, Helsinki, Finland, August 1985.

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

Kuhn, T.S. (1962). *The structure of scientific revolutions.* Chicago: University of Chicago Press.

Kuhn, D., & Phelps, E. (1982). The development of problem solving strategies. In H.W. Reese (Ed.), *Advances in child development and behavior.* New York: Academic Press.

Kulkarni, D., & Simon, H.A. (1987). The processes of scientific discovery: The strategy of experimentation. Unpublished working paper, Department of Computer Science, Carnegie-Mellon University, January 1987.

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

Larkin, J.H., McDermott, J., Simon, D.P., & Simon, H.A. (1980). Expert and novice performance in solving physics problems. *Science, 208*, 1335-1342.

Lenat, D. (1977). On automated scientific theory formation: A case study using the AM program. In J.E. Hayes, D. Michie, & L. Mikulich (Eds.), *Machine Intelligence 9*. New York: Halstead Press.

Levine, M. (1966). Hypothesis behavior by humans during discrimination learning. *Journal of Experimental Psychology, 71*, 331-338.

Mahoney, M.J., & DeMonbruen, B.G. (1977). Psychology of the scientist: An analysis of problem-solving bias. *Cognitive Therapy and Research, 1*(3), 229-238.

McCloskey, M. (1983). Naive theories of motion. In D. Gentner & A.L. Stevens (Ed.), *Mental Models.* Hillsdale, NJ: Erlbaum,

Medin, D.L., & Smith, E.E. (1984). Concepts and concept formation. *Annual Review of Psychology, 35,* 113-138.

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

Mitchell, T.M. (1979). An analysis of generalization as a search problem. *IJCAI, 6,* 577-582.

Mitchell, T.M., Utgoff, P.E., & Banerji, R.B. (1983). Learning by experimentation: Acquiring and refining problem solving heuristics. In R.S. Michaliski, J.G. Carbonell, & T.M. Mitchell (Eds.), *Machine Learning: An artificial intelligence approach, Volume I.* Palo Alto, CA: Tioga.

Mitroff, I.I. (1974). *The subjective side of science.* New York: Elsevier.

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.

Mynatt, C.R., Doherty, M.E., & Tweney, R.D. (1978). Consequences of confirmation and disconfirmation in a simulated research environment. *Quarterly Journal of Experimental Psychology, 30,* 395-406.

Neimark, E.D., & Santa, J.L. (1975). Thinking and concept attainment. *Annual Review of Psychology, 26,* 173-205.

Newell, A., Shaw, J.C., & Simon, H.A. (1958). Elements of a theory of human problem solving. *Psychological Review, 65,* 151-166.

O'Brien, D.P., Costa, G., & Overton, W.F. (1986). Evaluations of causal and conditional hypotheses. *The Quarterly Journal of Experimental Psychology, 38A,* 493-512.

Popper, R. (1959). *The logic of scientific discovery.* London: Hutchinson.

Rasmussen, J. (1981). Models of mental strategies in process plant diagnosis. In J. Rasmussen, & W.B. Rouse (Eds.), *Human detection and diagnosis of system failures.* New York: Plenum Press. NATO Conference Series.

Restle, F., & Greeno, J.G. (1970). *Introduction to mathematical psychology.* Reading, MA: Addison-Wesley.

Robinson, L.B., & Hastie, R. (1985). Revision of beliefs when a hypothesis is eliminated from consideration. *Journal of Experimental Psychology: Human Perception and Performance, 11*(4), 443-456.

Ross, B.H. (1984). Remindings and their effects in learning a cognitive skill. *Cognitive Psychology, 16,* 371-416.

Shepard, R.N., Hovland, C.I., & Jenkins, H.M. (1961). Learning and memorization of classifications. *Psychological Monographs,* Vol. *75*(13).

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

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

Siegler, R.S. (1987). The perils of averaging data over strategies: An example from children's addition. *Journal of Experimental Psychology: General,* . in press.

Siegler, R.S., & Liebert, R.M. (1975). Acquisition of formal scientific reasoning by 10- and 13-year-olds: Designing a factorial experiment. *Developmental Psychology, 11,* 401-402.

Simon, H.A. (1975). The functional equivalence of problem-solving skills. *Cognitive Psychology, 7,* 268-288.

Simon, H.A. (1977). *Models of discovery.* Dordrecht-Holland: D. Reidel Publishing Co.

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

Stavy, R., Strauss, S., Orpaz. N., & Carmi, G. (1982). U-Shaped behavioral growth in ratio comparisons. In S. Strauss (Ed.), *U-Shaped behavioral growth.* New York: Academic Press,

Tukey, D.D. (1986). A philosophical and empirical analysis of subjects' modes of inquiry in Wason's 2-4-6 task. *The Quarterly Journal of Experimental Psychology, 38A,* 5-33.

Tweney, R.D., Doherty, M.E., Worner, W.J., Pliske, D.B., Mynatt, C.R., Gross, K.A., Arkkelin, D.L. (1980). Strategies of rule discovery in an inference task. *Quarterly Journal of Experimental Psychology, 32,* 109-123.

Wallas, G. (1926). *The art of thought.* New York: Harcourt Brace Javanovich.

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

Wason, P.C. (1962). Reply to Wetherick. *Quarterly Journal of Experimental Psychology, 14,* 250.

Wertheimer, M. (1945). *Productive Thinking.* New York: Harper & Brothers.

Wetherick, N.E. (1962). Eliminative and enumerative behaviour in a conceptual task. *Quarterly Journal of Experimental Psychology, 14,* 246-249.

Whitman, J.R., & Garner, W.R. (1963). Concept learning as a function of form of internal structure. *Journal of Verbal Learning and Verbal Behavior, 2,* 195-202.

Table 1: Example of a complete protocol. CLR and GO commands have been deleted. BigTrak's behavior is shown in boldface type.

002 EXP: SO HOW DO YOU THINK IT MIGHT WORK?
003 Uh... it would repeat all of the steps before it, however many times
004 I told it to repeat it.
005 EXP: WELL... SO START WORKING ON IT NOW THEN.
006 Ok.
007 EXP: AND TELL ME EVERYTHING THAT'S GOING ON IN YOUR MIND.
008 Ok.
009 EXP: NOW PRESS CLEAR.
010 Ok, um... I'm gonna make it go
011 forward two, and then I'm gonna make it repeat that twice.

012 00:30 ↑ 2 RPT 2

↑ 4

013 Maybe not, uh.... ... repeat once

014 02:00 ↑ 1 ← 15 ↑ 1 HOLD 2 RPT 1

↑ 1 ← 15 ↑ 1 HOLD 4

015 Hmm.. guess that was it.
016 EXP: SO WHAT ARE YOU THINKING?
017 Um.. actually I have no idea now.
018 I'm trying to figure out what it is.
019 Um.. maybe it repeats the last step.
020 Ok, I'm gonna try that. ...... repeat once.

021 03:30 ↑ 2 ← 30 RPT 1

↑ 2 ← 60

022 All right, that backs up my theory.
023 Let me see if I can somehow make sure that that's what it does
024 is repeats the last step however many times that I tell it to,
025 so I'm gonna ... repeat it four times...

026 04:00 ↑ 2 ← 30 RPT 4

↑ 2 ← 30 ↑ 2 ← 30

027
028 That was strange, hmm... um... let me see that again.

029 04:30 ↑ 2 ← 30 RPT 4

↑ 2 ← 30 ↑ 2 ← 30

030 Ok, maybe it means repeat the last number...
031 however many steps before it that I put in,
032 that'll be the number after the repeat. For instance,
033 if I put repeat two, it'll repeat the last two steps.

If I put repeat five,it'll repeat the last five steps,
and if there's too many...
if the five is more than the number of steps in the program,
it'll just end it at whatever number of steps in the program,
so . . . repeat one, no, repeat two.

06:00 ↑ 2 ← 15 ↑ 2 FIRE 3 RPT 2

↑ 2 ← 15 ↑ 2 FIRE 3 ↑ 2 FIRE 3

All right, I think I might have gotten it.

06:30 ↑ 2 ← 15 ↑ 2 FIRE 3 RPT 3

↑ 2 ← 15 ↑ 2 FIRE 3 ← 15 ↑ 2 FIRE 3

Ok, I think I've gotten it. I'm gonna make it repeat four times.
. . . wanna repeat four...

07:30 ↑ 2 ← 15 ↑ 2 FIRE 3 RPT 4

↑ 2 ← 15 ↑ 2 FIRE 3 ↑ 2 ← 15 ↑ 2 FIRE 3

Ok, now I'm trying to figure out which order the repeat step goes.
If it does the first part of the program or if it does...if it starts
from the last part of the program, where repeat...
if I say repeat one, does it repeat the first step in the program,
or does it repeat the last step I pressed in? Um...repeat that
step...

09:00 ↑ 2 ← 15 ↑ 2 FIRE 3 RPT 1

↑ 2 ← 15 ↑ 2 FIRE 6

It goes from the last step,
and I don't understand why it doesn't go backwards.
Maybe it counts back two steps.
If I put repeat two, it would count back two steps,
starting from there and go until the last step. Alright,
...um...the last two steps were forward two and fire three,
so let me try and repeat that again.

10:00 ↑ 2 ← 15 ↑ 2 FIRE 3 RPT 2

↑ 2 ← 15 ↑ 2 FIRE 3 ↑ 2 FIRE 3

All right, now if I ... repeat five...
so if I put repeat four, it should do the whole program over again.
11:00 ↑ 2 ← 15 ↑ 2 FIRE 3 RPT 4

↑ 2 ← 15 ↑ 2 FIRE 3 ↑ 2 ← 15 ↑ 2 FIRE 3

Well, I think I figured out what it does.
EXP: SO HOW DOES IT WORK?

069 Ok, when you press the repeat key and then the number,
070 it comes back that many steps and then starts from there
071 and goes up to, uh...it proceeds up to the end of the program
072 and then it hits the repeat function again.
073 It can't go through it twice.
074 ......
075 EXP: GREAT.

Table 2: Common hypotheses and percentage of experiments conducted under each for Studies 1 and 2. Frequency of mention of each in hypothesis-search phase of Study 2

HYPOTHESIS[4]	% EXPERIMENTS UNDER EACH HYPOTHESIS		FREQUENCY OF MENTION
	Study 1	Study 2	Study 2
HS1: One repeat of last N instructions.	02	0	5
HS2: One repeat of first N instructions.	04	0	1
HS3: One repeat of the Nth instruction.	03	05	5
HN1: One repeat of entire program.	06	11	1
HN2: One repeat of the last instruction.	04	09	2
HC1: N repeats of entire program.	14	13	5
HC2: N repeats of the last instruction.	20	26	9
HC3: N repeats of subsequent steps.	02	0	3
Partially specified	03	0	1
Idiosyncratic	14	05	10
No Hypothesis	28	26	
	100	100	

[4] Hypotheses are labeled according to the role of N: HS - selector; HN - nil; HC - counter

Table 3: Attribute-value representation of fully-specified common hypotheses[5]

Rule	N-role	Rep-type	Bounds	# of reps	Prediction
HS1	selector	segment	last N	1	abcdCDef
HS2	selector	segment	first N	1	abcdABef
HS3	selector	instruction	Nth fm start	1	abcdBef
HN1*	nil	segment	all	1	abcdABCDef
HN2*	nil	instruction	prior	1	abcdDef
HC1	counter	segment	all	N	abcdABCDABCDef
HC2	counter	instruction	prior	N	abcdDDef
HC3	counter	segment	all following	N	abcdefEFEF

Test Program: abcdRPT2ef

[5] 1) * rules do not use N; 2) Uppercase letters in predictions show executions under control of RPT2; 3) Underlined letters reflect ambiguity in "repeat twice."

Table 4: Number of rejections/retentions of stated hypotheses, given confirming/disconfirming evidence, in Study 1

	Reject Confirm	**Retain Confirm**	**Reject Disconfirm**	**Retain Disconfirm**
All subjects	21	63	60	76
Experimenters	17	43	44	56
Theorists	4	20	16	20

Table 5: Performance summary of Experimenters and Theorists in Study 1

	Experimenters	Theorists	Combined
N	13	7	20
Time (minutes)	24.46	11.40	19.40
Experiments	18.38	9.29	15.20
Experiments with hypotheses	12.30	8.57	11.00
Experiments without hypotheses	6.08	0.76	4.2
Different hypotheses	4.92	3.86	4.55
Hypothesis switches	4.76	3.00	4.15
Experiment space verbalizations	5.85	0.86	4.10
$N\lambda$ combinations used	9.9	5.7	8.45

Table 6: Number of changes in program length or value of N in successive experiments for each group in Study 1

	N	λ	$N\lambda$	$\overline{N\lambda}$	**Total changes**
Experimenters	91	40	50	45	226
Theorists	21	18	11	8	58
Combined	112	58	61	53	284

Table 7: Performance summary of Experimenters and Theorists in phase 2 of Study 2

	Theorists	Experimenters	Combined
N	4	6	10
Have programming experience	4	1	5
Stated HS1 in phase 1	4	1	5
Time (minutes)	3.3	8.2	6.2
Experiments	3.0	7.5	5.7
Experiments with hypotheses	2.0	5.7	4.2
Experiments without hypotheses	1.0	1.8	1.5
Different hypotheses	1.5	3.0	2.4
Hypothesis switches	1.5	3.0	2.4
Experiment-space verbalizations	1.0	2.2	1.7
Nλ combinations used	2.5	5.7	4.4

Table 8: Number of rejections/retention of stated hypotheses, given confirming/disconfirming evidence, in Study 2

	Reject Confirm	**Retain Confirm**	**Reject Disconfirm**	**Retain Disconfirm**
All subjects	5	8	16	8
Experimenters	5	7	10	7
Theorists	0	1	6	1

Figure Captions

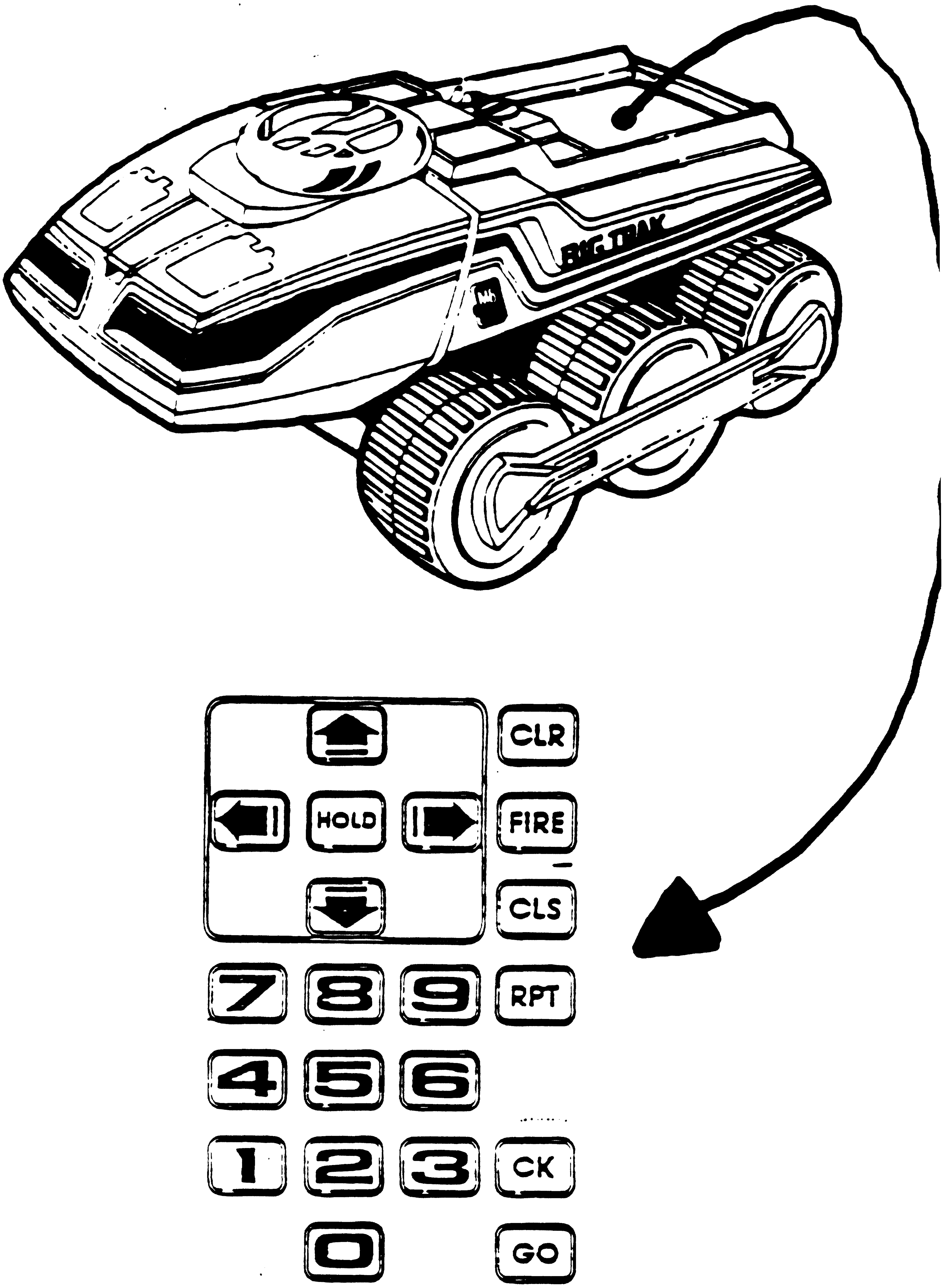
Figure 1: The BigTrak robot.

Figure 2: Frames for hypotheses about how **RPT N** works. Heavy borders correspond to common hypotheses from Table 2; 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.

Figure 3: Regions of the Experiment Space, showing illustrative programs and confirmation/disconfirmation for each common hypothesis. (Shown here is only the 10x10 subspace of the full 15x15 space.)

Figure 4: Process hierarchy for SDDS. All subprocesses connected by an arrow are executed in a sequential conjunctive fashion. All process names preceded by an asterisk include conditional tests for which subprocess to execute.

FIG 1



HYPOTHESIS SPACE

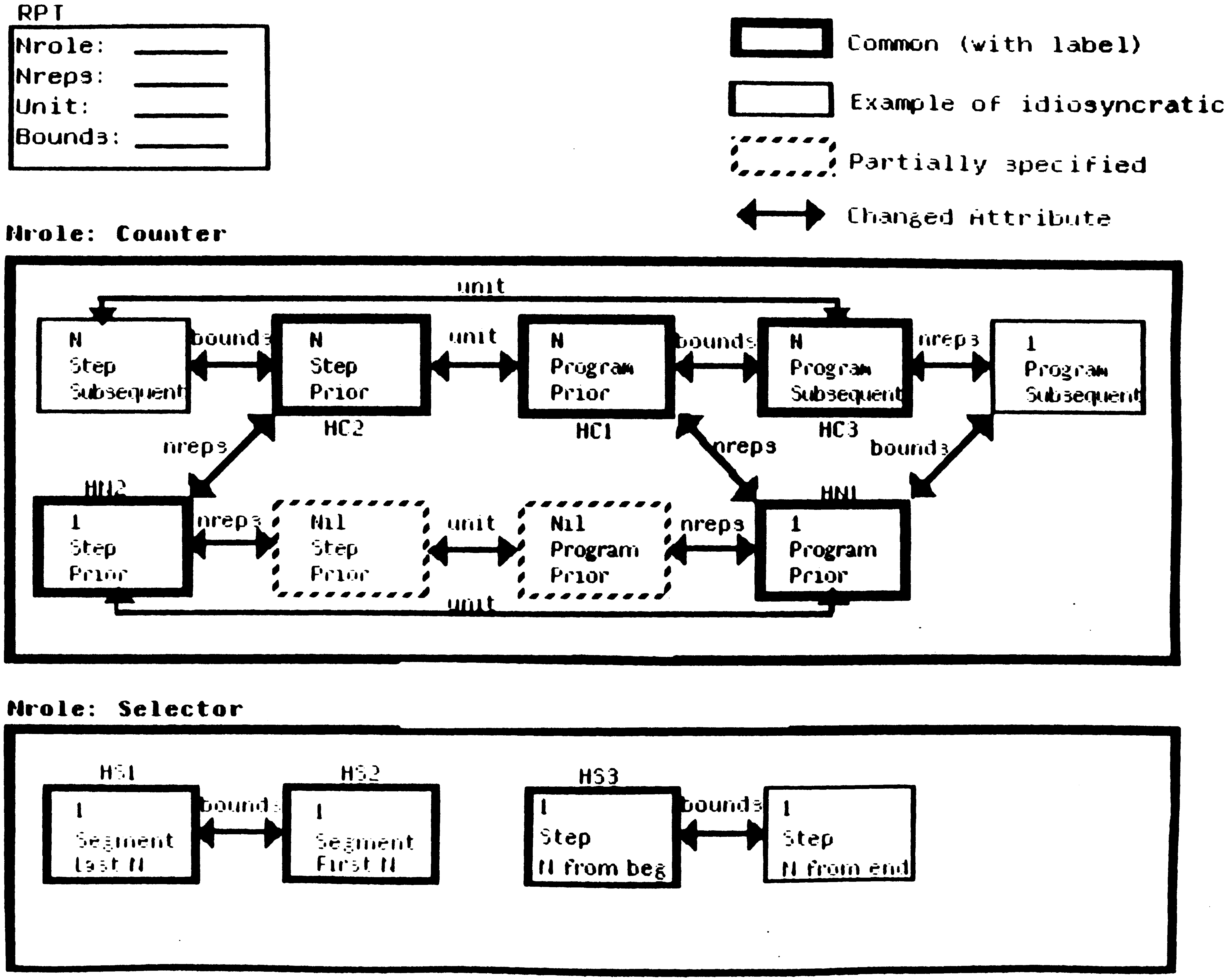
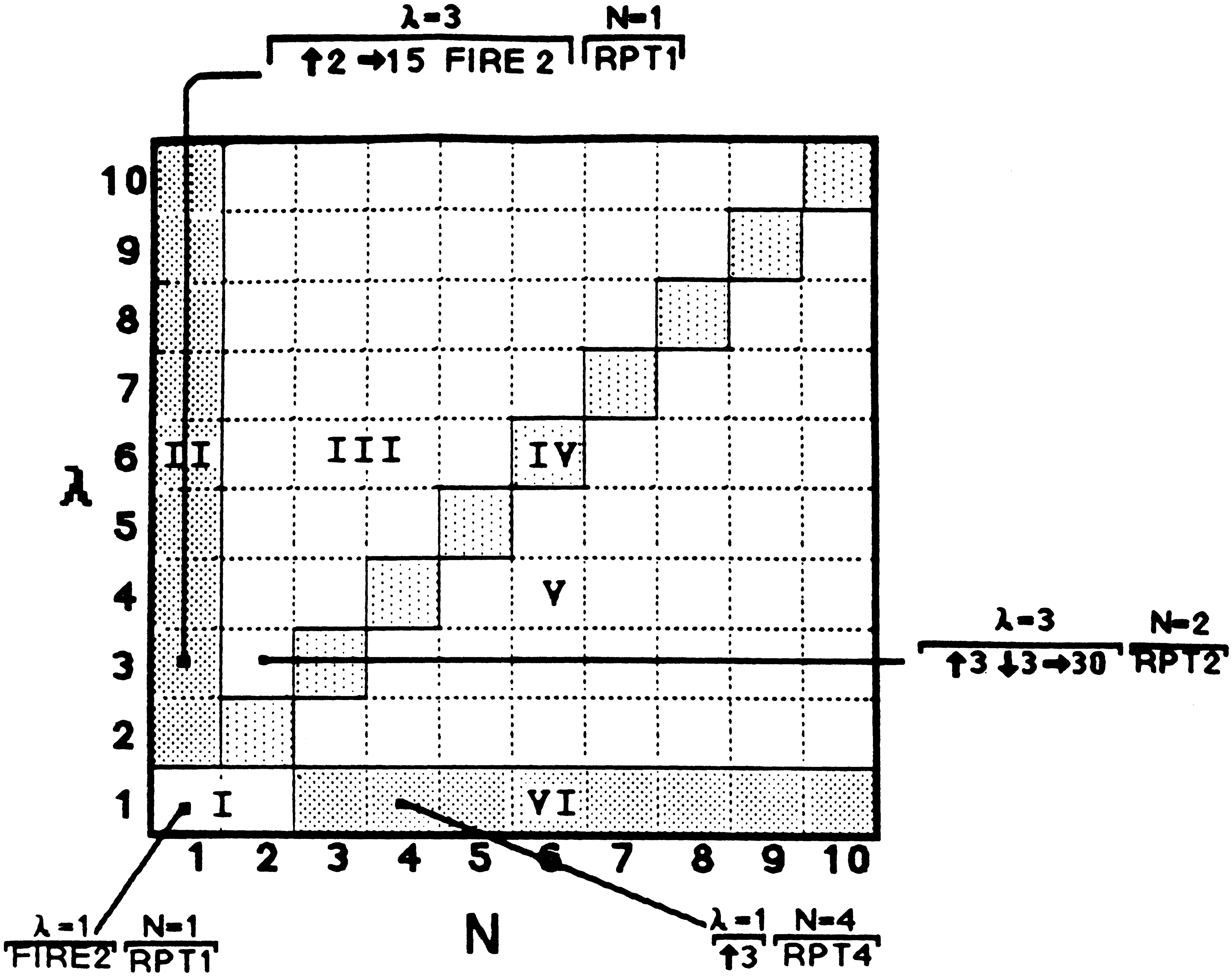


FIG 2

EXPERIMENT SPACE



		REGION					
		I	II	III	IV	V	VI
Hypothesis	HS1	+	+	+	+	+	+
	HS2	+	-	-	+	+	+
	HS3	+	-	-	-	-	+
	HN1	+	-	-	+	+	+
	HN2	+	+	-	-	-	+
	HC1	+	-	-	-	-	-
	HC2	+	+	-	-	-	-
	HC3	-	-	-	-	-	-

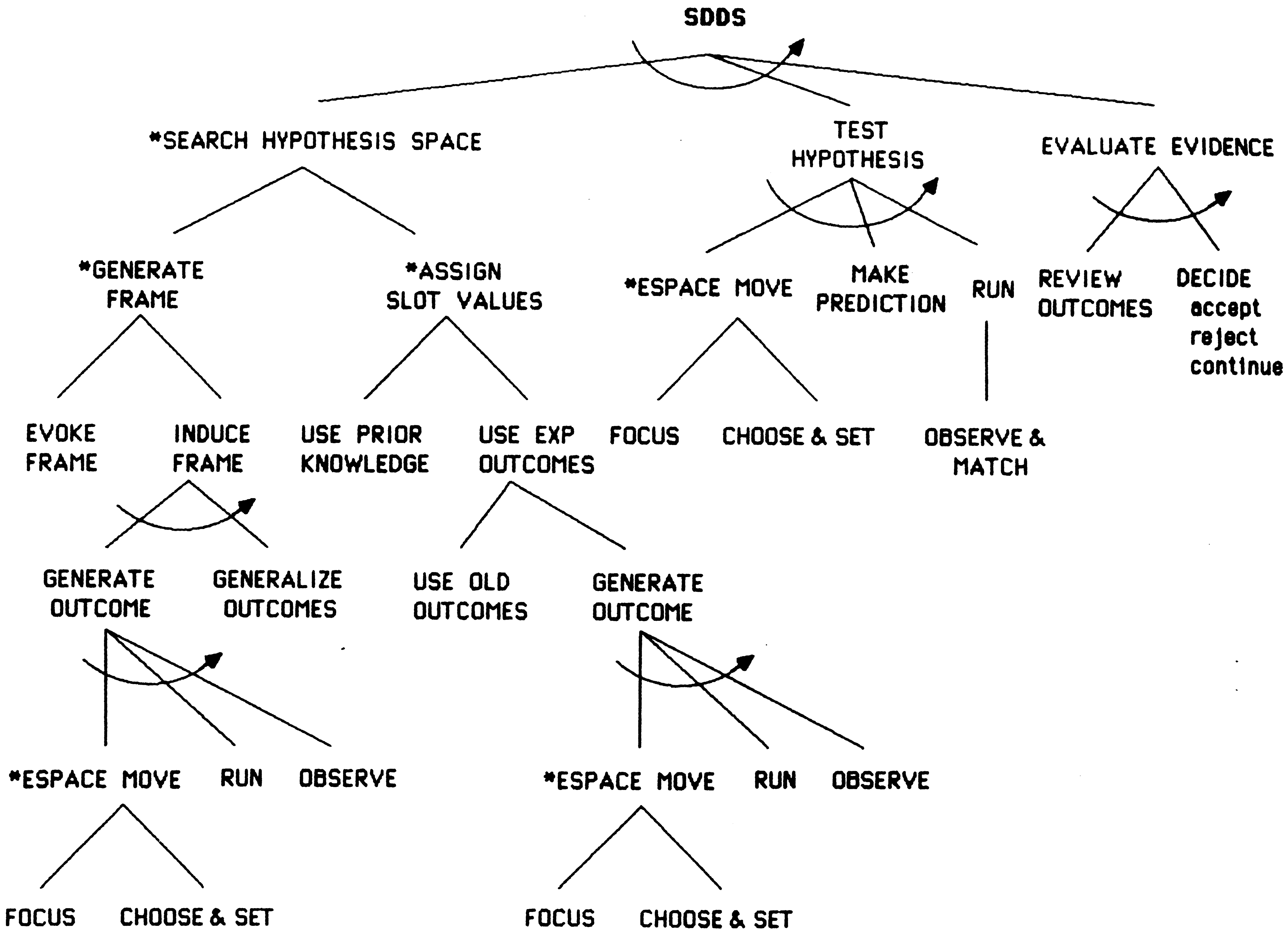

SDDS
*SEARCH HYPOTHESIS SPACE
TEST HYPOTHESIS
EVALUATE EVIDENCE
*GENERATE FRAME
*ASSIGN SLOT VALUES
*ESPACE MOVE
MAKE PREDICTION
RUN
REVIEW OUTCOMES
DECIDE
accept
reject
continue
EVOKE FRAME
INDUCE FRAME
USE PRIOR KNOWLEDGE
USE EXP OUTCOMES
FOCUS
CHOOSE & SET
OBSERVE & MATCH
GENERATE OUTCOME
GENERALIZE OUTCOMES
USE OLD OUTCOMES
GENERATE OUTCOME
*ESPACE MOVE
RUN
OBSERVE
*ESPACE MOVE
RUN
OBSERVE
FOCUS
CHOOSE & SET
FOCUS
CHOOSE & SET

Fig 4