# Modelling Scientific Discovery

## Thesis

# Modelling Scientific Discovery

Peter C-H. Cheng

BSc(Eng). MA.

Thesis submitted in fulfillment of the requirements for a
PhD in Artificial Intelligence
28th September 1990

Date of submission : 1st October 1990
Date of award : 23rd October 1990



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

## Summary

Traditionally, the Philosophy of Science has examined the nature of scientific discovery. In recent years, Cognitive Science has gathered together work in Artificial Intelligence (AI) and Cognitive Psychology that attempts to understand scientific discovery. However, at present, there is no generally accepted account of scientific discovery in any of these disciplines.

This thesis aims further to explore the nature of scientific discovery from an AI perspective, but does so within a clearly defined Framework, designed to structure cognitive science research on scientific discovery. The framework proposes a minimum set of components as a guide to the construction of acceptable accounts of scientific discovery. The focal concept is the Research Programme; a body of research that investigates a delimited set of phenomena using a Theoretical component and an Experimental component. The framework posits: three types of theoretical knowledge; three levels of experiments; inferences to apply and generate new theoretical & experimental knowledge; criteria for assessing the acceptability of theories & the reliability of experiments; and multiple levels of communication between the components.

Previous computer models and empirical studies of scientific discovery are reviewed. They tend not to offer complete accounts of scientific discovery, as defined by the framework. In particular, many completely ignore the crucial role of experiments.

The STERN computational model of scientific discovery is introduced. It instantiates all the components of the Framework. STERN currently models discoveries made by Galileo in the domain of naturally accelerated terrestrial motion, although it may be applied more generally. STERN has four main strategies that are used to make discoveries: (i) confirming existing hypotheses; (ii) generalizing experimental results to form new hypotheses; (iii) generating new hypotheses from known hypotheses; and (iv) generating new experiments.

STERN is more complete than previous computational models. As such it allows novel heuristics at the level of research programmes to be investigated and high level abilities to emerge from its complexity.

**For Jo**

who in her own way
made all this possible.

# Contents

# Preface

Science, in all its many and varied guises, is something that has always fascinated me. Its domination of my early adult education is a testament to this. More recently I have become interested in science at a meta-level and have examined the nature of science as a subject in itself. A brief excursion into the philosophy of science left me thinking that there must be a better ways to understand the nature of scientific discovery than just theoretical speculation based on logic and myths. Thus, I have found my way into AI and cognitive science. I now see the real hope of being able to treat scientific discovery, scientifically. This thesis is thus a distillation, or perhaps the culmination, of all of my past studies in science and of my studies of science.

Like all conceptual schemes, the ideas in this thesis have evolved over time. Early ideas that lead to the full development of the Framework for scientific discovery (presented in Chapter 1) can be found in Cheng (1989a & 1989b) and Cheng & Keane (1989a). Some previous speculations about the nature of Cognitive Science research on scientific discovery are given in Cheng (1990b). A prototype of STERN is to be be found in Cheng (1989b).

# Acknowledgements

My greatest thanks must go to Mark Keane. During all the stages of this research I could not have hoped for better supervision. His unfailing encouragement, highest academic standards, clarity of purpose and seemingly endless energy have altogether been an indispensable catalyst.

Thanks should also go to Marc Eisenstadt for all his support and excellent advice on this work and in so many other matters. Thanks too to all those who helped by commenting on and proof-reading versions of this thesis, and to everyone in HCRL and the OU who has helped to make my stay such an intellectually stimulating and enjoyable time.

I will be forever grateful to my parents for teaching me the real value of knowledge and for being completely unreserved in letting me pursue it.

This research was performed under a Science and Engineering Research Council studentship.

# List of Tables



continued

## List of Tables Continued

# List of Figures

# Chapter 1

## Introduction: Components Of Scientific Discovery

As a human intellectual endeavour science has been successful. It has enabled humanity to develop and expand its knowledge of the world, to explore the universe, to predict future events and to found new technologies that manipulate and control nature. However, the understanding which science affords us has not been reflexive; science has not lead to an understanding of science. There is no generally accepted view of how science really works. Philosophers of science have propounded radically different views ranging from models based on logical systems (e.g. Popper, 1959, 1965; Fiegl, 1970; Suppe, 1977), through to sociologically-oriented views which assume less logical foundations (e.g. Kuhn, 1970; Feyerabend, 1975).

More recently, an alternative avenue of investigation has opened up, with the first attempts in Artificial Intelligence (AI) and Cognitive Science to develop computer programs that do scientific discovery. Some of these attempts, as we will see in Chapter 2, have provided detailed simulations of famous discovery episodes. Others have provided programs that will make discoveries, albeit not in a manner that mirrors human abilities. However, just as in the Philosophy of Science, no general consensus has emerged about the fundamental nature of scientific discovery in Cognitive Science, although the concrete nature of the enterprise has provided a useful methodology for understanding the phenomenon. However, the roots of the lack of consensus in Cognitive Science are quite different to those in the Philosophy of Science. Artificial Intelligence research has tended to be *technique-driven*: that is, researchers have taken an AI technique (e.g. data space search using regularity

spotters) and tried to build discovery systems based on the technique (e.g. Langley et.al., 1987; Falkenhainer & Michalski, 1986).

The present thesis is very much within this Cognitive Science tradition with one important exception. Rather than be technique driven, I will attempt to propose a general framework for scientific discovery, which will then be realized in a subsequent set of programs. The idea is to have a more principled approach to AI work.

This *Framework For Scientific Discovery* specifies a set of components that seem important for the adequate characterization of scientific discovery. Like any framework, it should provide a clear conceptual foundation on which a greater understanding can be built. This chapter will lay out the framework. In chapter 2, I will show how useful the framework is in setting previous AI work in context and revealing its limitations. Then much of the remainder of the thesis realizes many components of the framework in a discovery program called STERN.

*A Framework For Scientific Discovery*

The framework views the scientific enterprise as consisting of a single scientist or groups of scientists carrying out research programmes. Clearly, the nature of these programmes will depend on what is being investigated, the science in question, and prior theoretical and experimental developments in the field. However, the important abstraction to keep in mind is that a *scientific research programme* involves the investigation of a delimited set of phenomena using both theory and experimentation.

This may seem quite obvious but is less so when one considers that philosophers of science have only recently begun to acknowledge the role of experiment (Hacking, 1983; Galison, 1987; Franklin, 1988; Gooding *et.al.*, 1989) and that AI researchers typically do not model experimentation in their computer

programs (e.g. Langley *et.al.*, 1987; Fisher & Zytkow, forthcoming; Thagard, 1988a, 1989a; Rose, 1988a). However, we clearly need to specify what we mean by theory and experimentation.

Within the present framework *theory* or *theoretical knowledge* is characterized as sets of functions. These functions characterize the behaviour of a phenomenon (or event in the world). That is, they relate together the initial conditions and final states of the event. The function predicts or explains (accounts for) the way in which the initial state of the phenomenon changes into some final state. In characterizing *theoretical knowledge* the framework distinguishes between entities that differ in their generality or abstractness. At the most general level there are *hypotheses*, at the next level *models* and, finally, *instances* of these models. In addition to these aspects of theory the framework also assumes that there are criteria for determining the adequacy of theoretical knowledge (called *acceptability criteria*).

The second main aspect of the framework is *experimentation*. In the framework, experiments are characterized as "black boxes", with input parameters that constitute manipulated/controlled variables and output parameters that are the observations and measurements that a scientist makes. Again, as in the case of theory, experiments are defined at three levels of generality involving *experimental paradigms*, *experimental set-ups*, and *experimental tests*. Two important issues I will consider in this component are how new experiments are invented and how scientists establish the reliability of experimental results.

One of the final things that the framework is centrally concerned with, is the way in which the two main components of a research programme - theory and experiment - interact, correspond and communicate with one another.

Let us now consider the framework in more detail.

## 1.1 SCIENTIFIC RESEARCH PROGRAMMES

A scientific research programme involves the directed use of both theory and experiment to investigate a delimited set of phenomena. In a brand new research programme the phenomena are identified by pre-theoretic or even pre-scientific means. However, typically research programmes occur within an established field. In this case the phenomena may be specified as a topic within that field.

In this thesis individual scientists working more or less in isolation are typically considered. Thus the major role of research programmes, here, is to bring together the main theoretical and experimental components. (Research programmes will have an even more substantial role when cooperative and competitive investigations amongst many scientists are considered - multiple parallel research programmes will need to be modelled.)

## 1.2 THEORY

The theory component of the framework is composed of four sub-components. These four sub-components characterize the general nature of theoretical knowledge, the types or levels of knowledge, definitions of theoretical inferences, and acceptability criteria.

### 1.2.1 Theoretical Knowledge

Theoretical knowledge attempts to explain or predict the behaviour of a phenomenon as it changes in a natural or experimental environment. The changes considered are variations to the characteristic conditions, properties and quantities (attributes in general) of the phenomenon. A set of values for these attributes at a specific time constitutes a state of the phenomenon. For example, we may be investigating the free fall of bodies under the effect of gravity, in which case magnitudes of quantities like height, speed and weight, and properties like the shape and material of the body and the medium it travels through, may help to define the state at a given time.

Given the characterization of phenomena in a state-based way, the nature of theoretical knowledge can be conceptualized as a *state transformation function*, expressed mathematically as,

$$T\{S_1\} = S_2, \qquad \ldots (1.1)$$

where $S_1$ is the initial state of the phenomenon defined in terms of specific values of the characteristic attributes, and $S_2$ is the final state with similar attributes but different values. $T$ is the transformation function: a mathematical or propositional formalism that states how $S_1$ is related to $S_2$. Simply, T describes, predicts or explains how the initial values of the attributes change into their final values.

Consider Galileo's law of free fall as an example. The attributes considered by the law are speed and height and they are related by the equation:

$$\text{speed} = c \,.\, \text{height}^{1/2}, \qquad \ldots (1.2)$$

where $c$ is a constant, say 10. Now, given that the magnitude of both *height* and *speed* are zero in the initial state, then for a final state in which the height (measured downwards) is 4 we may calculate that the speed is 20 (ie. $10.\sqrt{4}$). This is a rather trivial example, but it succinctly illustrates the transformation function idea.

### 1.2.2 Three Types Of Theoretical Knowledge

Given the above general formulation of theory it is important to distinguish between types of theoretical knowledge with a view to understanding the complexities of theory in scientific discovery. These three types of theoretical knowledge of are; hypotheses, models and instances.

*Hypotheses* are the most general type of theoretical knowledge. Their state transformation functions attempt to be universal accounts for all relevant manifestations of the phenomenon in the differing situations defined by the research programme. For example, we may attempt to apply Galileo's law not only to bodies in free fall, but to swinging pendulums, projectiles, and balls rolling down ramps.

That is, to all naturally accelerated terrestrial bodies.

*Models* are a more specific type of theoretical knowledge than hypotheses. They attempt only to account for the phenomenon in one of the many situations defined by the research programme. A model's state transformation function is not expected to be applicable to other situations. For example, we may use the "law of free fall" hypothesis to infer the shape of the path described by projectiles as they fly through the air. The function defining the trajectory is a *model*, and as such has no relevance to the explanation of other situations; e.g. the motion of balls as they roll down ramps.

The *instance* type of theoretical knowledge is even less general than a model. An instance comprises a series of states of the phenomenon in just one situation and relates to a single event. In other words the values of the attributes characterizing the phenomena are specified before, during and after the event. For example the prediction that a ball rolling down a ramp will have travelled 10, 40, and 90 cm after 0.1, 0.2 and 0.3 seconds respectively, when the inclination of the ramp is 2°, is an instance. A *term* is a variable standing for some attribute, for example distance or inclination, that may be assigned a specific value at a particular moment. When one term is varied so that values of another may be calculated, they are called the *independent* ($S_1$) and *dependent* ($S_2$) terms, respectively.

The relationship between the three types of knowledge is one of partial instantiation; that is, models are more specific manifestations of hypotheses and instances are specific versions of models. Exactly why the instantiation is partial will be seen later (in Chapter 4).

### 1.2.3 Origins Of Theories And The Nature Of Theoretical Inferences

The nature and types of state transformation functions have been considered, but how they come into being and what use they are put to needs to be discussed. For example, new functions can be generalized from data or generated from existing functions, and known functions allow predictions to be made or explanations to be

given. Consider the genesis of transformation functions first.

The inference to new state transformation functions may take many different forms but two common ones are (i) the generalization from data, and (ii) generation from one or more existing functions by modification or combination. Finding a two term mathematical function (like equation 1.2) for a series of related Cartesian coordinates is an example of the generalization to a transformation function from data. As an example of the second form of this type of inference, consider a scientist whose is given the following two laws that describe different aspects of the same phenomenon using equations that refer to theoretical terms:

$$A.B = C \qquad ...(1.3a)$$

and

$$A.D = E. \qquad ...(1.3b)$$

The generation of a new function by combining the two equations may yield, as one of many possibilities, the following:

$$A.B.D = C.E\,. \qquad ...(1.3c)$$

Understanding how and why scientists make certain types of modifications or combinations of functions is of course part of the investigation of the nature of scientific discovery. Other forms of new function generation may require the use of background knowledge or even the borrowing of formalisms from other research programmes by the use of analogy.

Like the genesis of state transformation functions, the different ways in which they are used or applied are many and varied. Consider just two forms of inference. (i) *Prediction* occurs when the initial state of the phenomenon is given and the function is used to work out what the final state will be. For example, knowing equation (1.2) it is possible to predict that the speed after a fall of 4 metres from rest will be 20 m/s. (ii) *Explanation* requires knowledge of both the initial and final states, and the function is used to demonstrate or elucidate the way in which they are related. For example, concluding that the speed of a body falling from rest is 20

m/s after 4 metres is explained by Equation (4.2), which tells us that speed is in proportion to the square root of height.

The generation of new state transformation functions and their application to phenomenon by prediction and explanation is a central part of the theoretical side of scientific discovery.

### 1.2.4 Acceptability Criteria For Theories

Assessing the acceptability of theoretical knowledge is another integral part of this component of scientific discovery. In the framework, I assume that the main, but not the only, index of the acceptability of a theory is the number or range of different phenomena to which it is applicable (ie. which it successfully predicts or adequately explains). This has variously been referred to as the *explanatory breadth* (Thagard, 1989a), *consilience* (Thagard, 1988a), or *predictive scope* (McAllister, 1989). Other acceptability criteria to consider are pragmatic considerations in the development of theories, and in particular, the *tractability* or ease with which inferences can be made from a theory.

## 1.3 EXPERIMENT

The treatment of experimentation within a research framework can be broken up in much the same way as the treatment of theoretical knowledge. A general abstract conception of experiments is given, that is then broken up into a three-levelled scheme which parallels the hypothesis, model and instance levels of the theory component. There are also related issues in this component about the genesis of experiments and their reliability.

### 1.3.1 The Structure Of Experiments

Under the framework experiment is viewed as a mechanism that treats a phenomenon as a "black box". The scientist investigates the underlying nature of the phenomenon via a set of specified inputs and outputs. The inputs attempt to

control some aspects of the phenomenon and manipulate others, whilst the outputs reveal the changes that result from these particular inputs.

In an experiment, a phenomenon is instantiated in a manner that allows input parameters (Inputs–M) to be manipulated and output parameters (Outputs) to be measured or observed. Some input parameters are fixed (Inputs–C), their values held constant to tightly control the experimental environment. The form of experiments can thus be represented by the equation:

$$E\ (\text{Inputs-M/Inputs-C}) = \text{Outputs} \qquad \ldots\ (1.4)$$

where the phenomenon in the black box determines the hidden functional relation, *E*, between Inputs-M/Inputs-C and Outputs. In this scheme, the experimental apparatus is required to instantiate and manipulate the phenomenon, and instruments are needed for measurement and observation. This conception of experimentation is depicted diagrammatically in Figure 1.1. Ideally just one Input–M should be manipulated at a time when performing an experiment to prevent ambiguity over the extent to which a parameter affects the phenomenon.

The relationship between theories as state transformation functions and experiments as black boxes (i.e. the similarity between equations 1.1 and 1.4) is at the heart of the framework. The Independent and Dependent theoretical terms represent experimental Input-M and Output parameters, respectively. The hidden experimental relationship, E, is of course the thing that remains to be discovered and characterized by theoretical state transformation functions. Hence, under the framework, the aim of scientific discovery can be characterized as the finding of state transformation functions that are ever closer approximations to the underlying nature of the phenomenon.

For example, in one experiment Galileo rolled balls down a ramp. The Input-M parameter was the distance down the ramp and examples of the Input-C parameters were the inclination of the ramp, and the weight and size of the ball. The Output, time, was measured using a water clock. Hence, the underlying nature of the

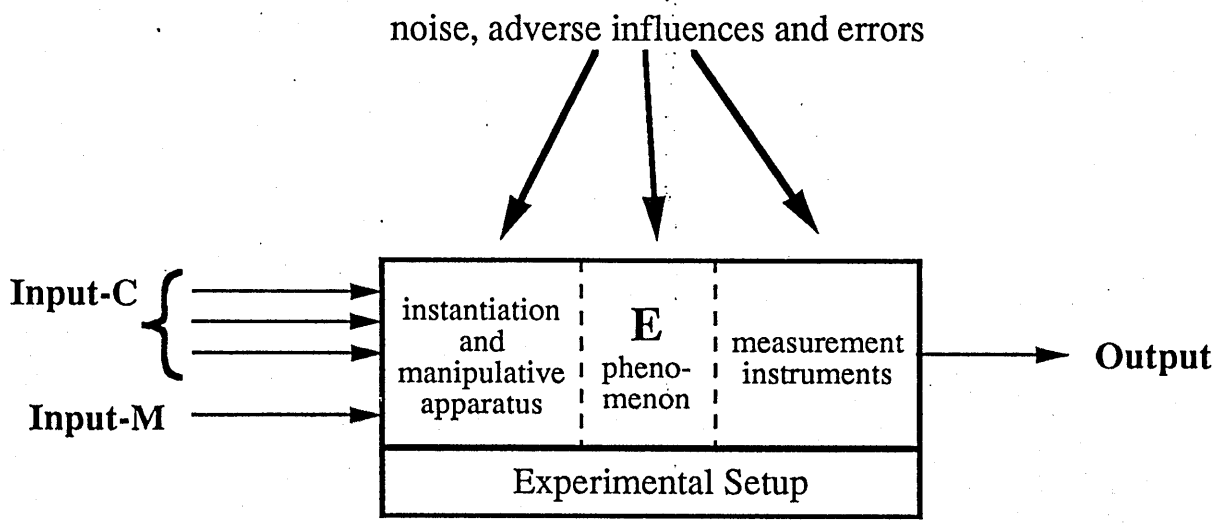



**Figure 1.1 Black Box Conceptualisation Of Experiments**

phenomenon was described by the "law of free fall" state transformation function.

The majority of experiments can be characterized in a similar way. The exceptions are cases where manipulative control is absent, because the phenomenon only occurs naturally and cannot be instantiated in an experimental environment. In such cases *mere observation* occurs, that yield results that are much less reliable than those formed in "normal" experiments. Examples of fields in which this typically occurs are astronomy, geology and areas of biology.

### 1.3.2 The Levels Of Experiments

More specifically, the above general characterization of experiment is treated at three levels of generality in the framework: as experimental paradigms, experimental setups, and experimental tests.

At the most general level, within most sciences there are distinguishable classes of experimental situations, which are used to investigate the phenomena within a research programme. These classes of experiments are called *experimental paradigms*. An example of one of Galileo's experimental paradigms was the inclined plane paradigm, in which balls were rolled down a smooth straight ramp. This constitutes one class of experiments, because many different configurations of the apparatus and entities can be used to perform different tests.

At a more specific level there are *experimental setups*. These are instantiated experimental paradigms, that is manufactured experimental apparatus and equipment for manipulating input parameters and instruments for observing and measuring output parameters. Galileo considered several experimental setups given the inclined plane experimental paradigm. One setup used the inclined plane to investigate the length of time for the ball to roll different distances with the ramp at a constant inclination. Another examined the relation between speed and inclination for a fixed height. Thus, different experimental setups provide different means of instantiating, manipulating and observing the phenomenon with resultant variations of the input and output parameters.

Finally, at the most detailed level one has specific *experimental tests*. An experimental test refers to a particular experimental trial on a specific manifestation of the phenomenon. In an experimental test an experimental setup is selected and particular parameters are chosen to be the Input-M, Input-Cs and Output parameters. The experiment is then performed with a series of input values for which output values are recorded. As such, experimental tests are partial instantiations of experimental setups, which in turn are partial instantiations of experimental paradigms. The exact sense in which the relations hold will become clear in subsequent chapters when we deal with these components in detail.

### 1.3.3 Experimental Processes

Given this view of three levels of experiment, there are two main sets of processes needed to complete the picture.

One concerns those processes required to carry out an experiment. These processes involve (i) the selection of a particular experimental paradigm given an appropriate manifestation of the phenomenon being investigated and (ii) the instantiation of this paradigm in an experimental setup that permits one to carry out experimental tests. Clearly, when a theory is being tested these processes are carried out with close reference to appropriate theoretical knowledge (as we shall see later).

The other type of process concerns the genesis of experiments. All the possible experimental paradigms in a research programme do not simply exist but must be conceived and manufactured by scientists. Experimental paradigms may already exist in another domain that a scientist may simply borrow and use in the domain of interest. However, scientists must also sometimes invent new experiments to carry out or continue a research program. I will not provide a full account of how new experiments are invented, since this is very much an open question. However, I will consider one method that has been used by scientists in the past, which involves the novel combination of two existing experimental paradigms to form a

new paradigm. Other methods are left to future investigations and some no doubt are yet to be discovered.

### 1.3.4 Reliability Of Experiments

One of the important cornerstones of acceptable theoretical knowledge is the acceptability or trustworthiness of the experimental evidence on which the theory is based. Hence, there are important techniques for determining the reliability of experimental results. Specifically, it is essential to ensure that the input manipulations (inputs-M) are solely responsible for the changes to the phenomenon measured in the outputs. In terms of the black box conception of experimentation, other influences such as noise, artifactual events and errors, may affect the phenomenon rather than the Inputs-M, or may cause false readings in instruments measuring the outputs. The strategies to deal with such potential sources of errors have recently begun to be considered by philosophers of science (Hacking, 1983; Galison, 1987; Franklin, 1988; Gooding *et.al.*, 1989) and no complete model of scientific discovery can ignore them. We will see some of the techniques that have been used to overcome background noise, one common type of extraneous variable in experiments.

*A Convenient Representation for the Structure Of Scientific Knowledge*

Since research programmes are constituted by the combination of theory and experiment and the three types of theory and three levels experiment are related by partial instantiation, the framework can conveniently be represented as the hierarchy shown in Figure 1.2. However, for completeness, we also need some indication of the fact that the theory and experiment components communicate in various ways with one another.

## 1.4 COMMUNICATION BETWEEN THEORY AND EXPERIMENT

The *communication* of the theory and experiment components is a central part of the process of scientific discovery. As theory and experiment each have three

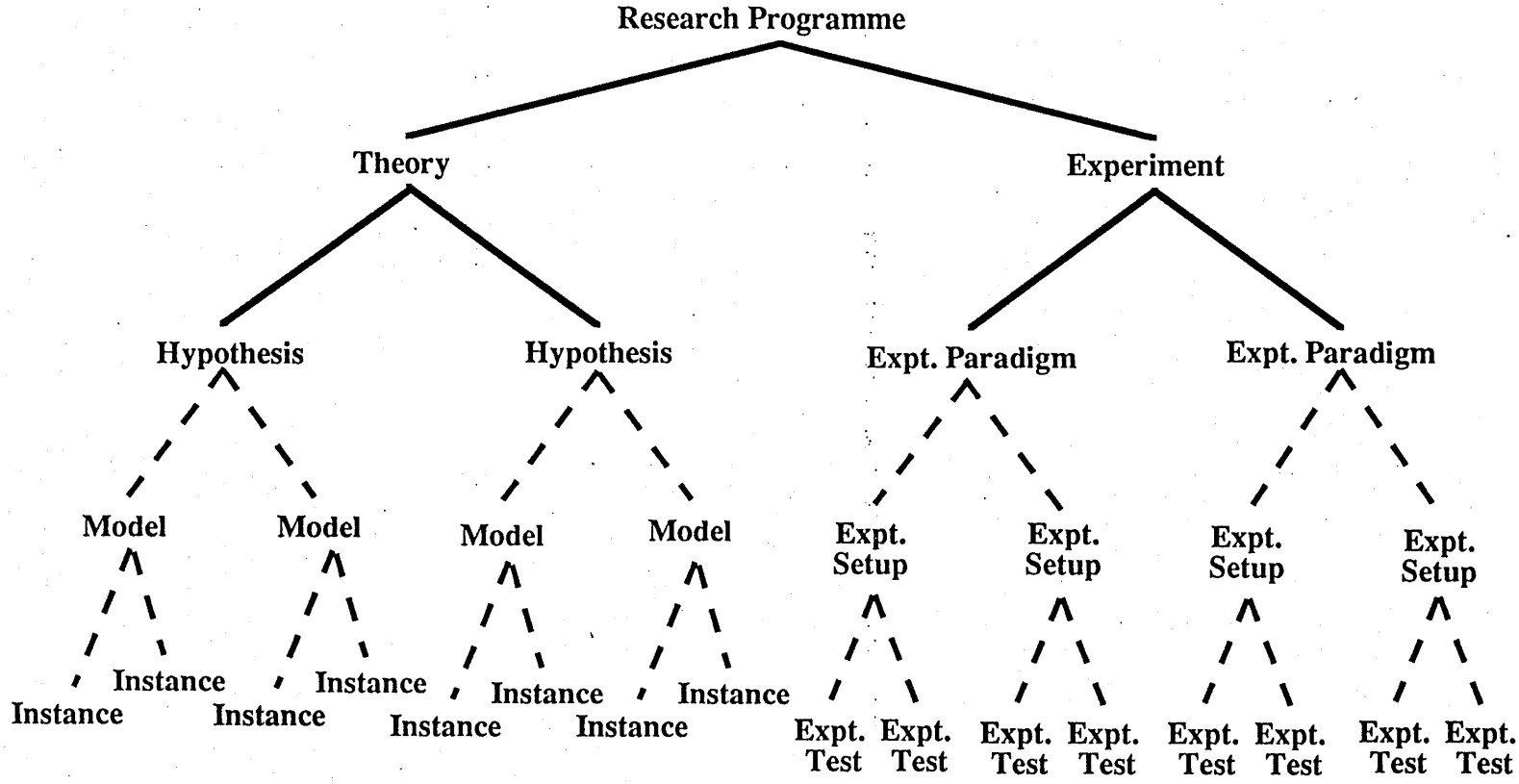



**Figure 1.2 Scientific Knowledge Hierarchy**

Key: reading downwards solid lines are read as "has part", and dotted lines "generates or generalized from"

different types of component there is substantial scope for many different levels and types of information transfer.

The most direct connection between theory and experiment occurs at the most specific level of each component. As mentioned above, the terms of instances in the theory component directly represent experimental parameters in experimental tests. Such terms and parameters may *correspond* to each other only if they are of the same type; that is, theoretical terms for quantities like time and distance refer directly to experimental parameters that measure time and distance. This acknowledges that quantitative physical scales of measurement, founded on base units (such as the metre, kilogram, second, Ampere etc.) defined by international convention, are universally adopted throughout the physical sciences (at least). The simplifying consequences of this view are that no special interpretation is needed to recognize a parameter given a term, or vice versa, and that numerical values of corresponding entities have the same magnitude. The role of this level of correspondence will be seen in the chapters below, along with many other types of communication between the two sides of scientific research programmes.

## 1.5 FRAMEWORK SUMMARY, ORIGINS AND SCOPE

Figure 1.3 shows a graphical summary of the components of the framework. The framework permits a high level description of scientific discovery to be stated:
*In a scientific research programme a delimited set of phenomena is investigated by experiment and characterized by theory. Experiment treats phenomena as black-boxes, manipulating the input parameters to observe the effects on output parameters. Theoretical knowledge consists of state transformation functions that account for the hidden contents of the black boxes, inferred by the direct correspondence between theoretical instance terms and experimental test parameters. At a more general level, models and hypotheses are formed, accounting for the phenomena across different experimental setups and paradigms.*

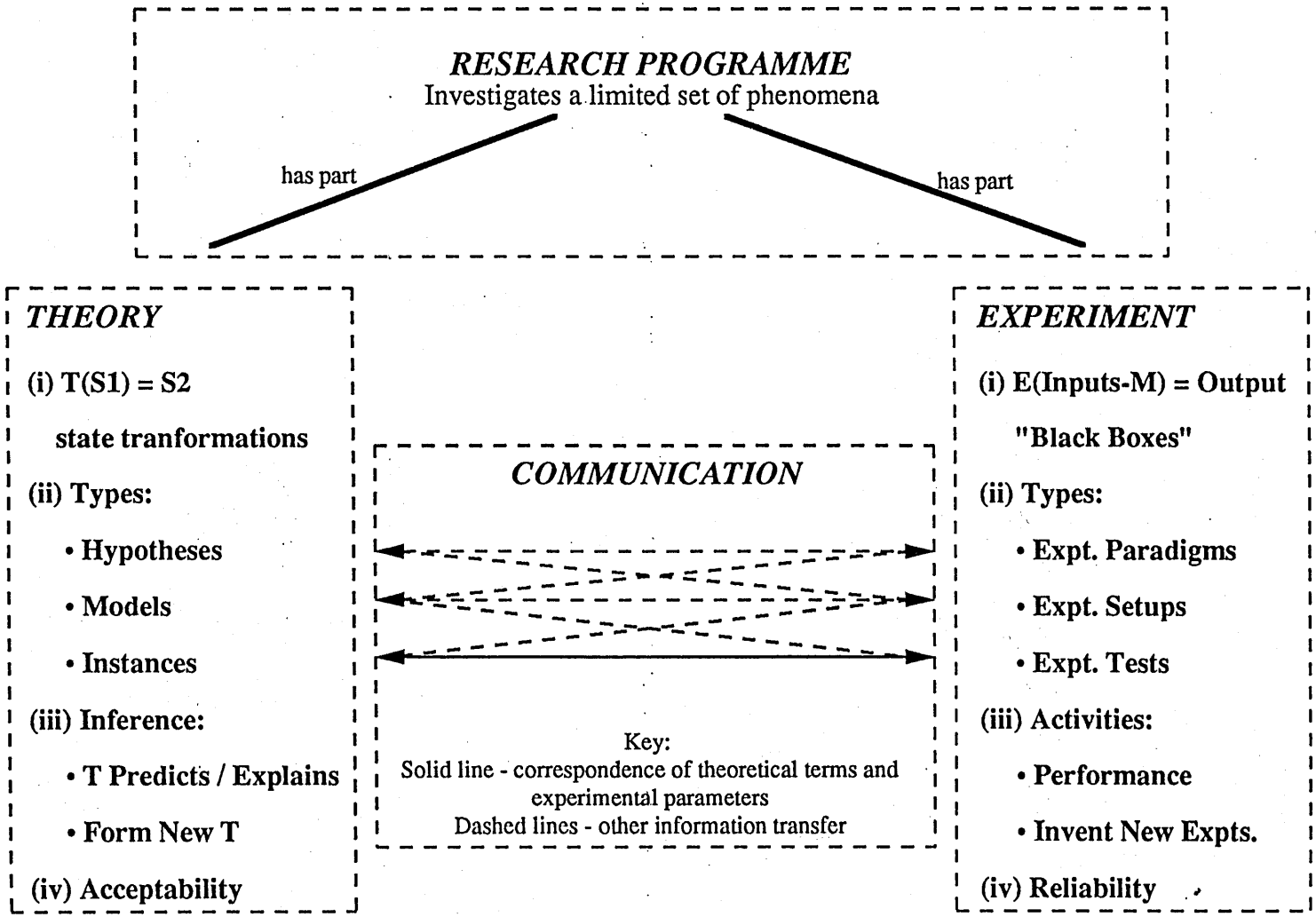



Figure 1.3 Summary Of The Framework For Scientific Discovery

*This theoretical knowledge is assessed by acceptability criteria, which is influenced by the reliability of the experiments carried out.*

The framework is composed of a number of different concepts that have been moulded into an integrated whole. The sources of the ideas are many and varied, but some can be more clearly identified than others. The concept of research programmes in science is one that is well established in the philosophy of science (e.g. Lakatos, 1974). Research programmes are central to Lakatos's view of the methodology of science. However, research programmes in the framework are quite different in that they are defined not only in terms of the theoretical knowledge that is present but also with regard to the available experiments. Treating experiments as black boxes has its origins in Cheng (1988) that uses this conceptualization to compare the strategies used by experimenters with concepts from modern engineering control theory. The importance of independent and dependent theoretical terms is widely acknowledged throughout science. However, the development of the framework's state transformation function view of theoretical knowledge was influenced, in a very general way, by Holland *et.al.*'s (1986) representation of theories as clusters of rules. The idea that there are types of theory and levels of experiments comes, in part, from the different levels of constraint that Galison (1987) sees in theories and experiments. Finally, the correspondence level of communication between theory and experiments is closely related to scales of measurement, defined in terms of base units, that are ubiquitous in science and engineering.

It is clear that the framework is quite a general characterization of the nature of science. The framework can be applied to a great range of important scientific fields. In particular, the quantitative physical sciences can be dealt with most directly by the framework - it is this type of scientific research that we will consider in this thesis. However, other fields of science may also be considered. For

instance, behaviourism in psychology seems likely to be amenable to a treatment by the framework. Nevertheless, we must mindful of potential exceptions to the framework. For example, Darwin's theory of evolution explains how species evolve by the environmental selection of individuals that differ due to random variations (the survival of the fittest). Further work will be required to show whether or not the Darwinian view of the evolutionary process can be expressed in terms of inputs and outputs to a state transformation function.

## 1.6 THESIS OVERVIEW

A framework for the characterization of scientific discovery has been introduced. The work throughout this thesis relies heavily on the framework. It is used: (i) to organize and analyse previous work; (ii) as the basis of a computational scientific discovery system; and (iii) to help address issues on the nature of research in cognitive science concerning scientific discovery.

### 1.6.1 Review Of Previous Work

There has been much previous work on scientific discovery in the fields of philosophy of science, cognitive psychology and AI. However, only the concrete work from cognitive science and AI is reviewed in this thesis (Chapter 2). Cognitive psychologists have performed empirical studies to investigate how humans perform particular scientific tasks and how they behave in simulated scientific discovery environments. Researchers in AI have built many computational systems that solve real scientific problems or that model episodes of discovery from the history of science. Work in both fields can be classified using four categories derived from the framework. The categories are defined in terms of which types of theoretical knowledge are given as initial conditions and what is done with that knowledge. The categories are: (i) the instance-driven formation of models; (ii) the hypothesis and model-driven generation of instances; (iii) assessing the acceptability of known hypothesis; and (iv) multiple-process systems that encompass all the

previous categories. We will see that all of this work is limited in various ways, but one major deficit stands out: they all ignore the crucial role of experiments in scientific discovery.

### 1.6.2 Galileo's Discoveries On Natural Motion

Galileo is perhaps the first *scientist* in the modern sense of the term, because he not only theorized about the nature of phenomena, but performed experiments in which phenomena could be manipulated and accurately measured. A major part of this thesis is concerned with the computational modelling of Galileo's discoveries in the domain of the naturally accelerated motion of terrestrial bodies. We will consider the experiments that Galileo used and the theories he postulated (in Chapter 3). Galileo manufactured experimental paradigms and used setups to design experimental tests. But more than that, he also invented novel experimental paradigms based on the ones he already had. Galileo initially assumed that laws originating from Aristotle were correct but soon found, by experimental testing, that they were unacceptable. Galileo then used various experiment-led and theory-led approaches to investigate natural motion. He obtained a deep understanding of the phenomenon in a qualitative manner, from which he eventually inferred his law of free fall. This law was tested using further experiments and shown to be the only generally acceptable hypothesis in the domain.

### 1.6.3 STERN The Discovery System

STERN (Scientific Theorist and Experimental Researcher, version N=0) is a computational model of scientific discovery (Chapter 4). It fully implements all of the aspects of the framework. STERN has a hierarchy of frames to instantiate the types of theory and the levels of experiment. Other frames represent theoretical terms and experimental parameters. This permits the modelling of communication between the theoretical and experimental components. The criteria for the acceptability of theories used in STERN is based on the relative success of instances

and models, as applied to experimental setups and paradigms.

STERN models the Galilean episode, so it is given knowledge representations specific to that domain. Equations and special qualitative representations are used. A sub-program is invoked by STERN to simulate the performance of experiments. The experimental results produced by the simulator contain levels of noise that are realistic compared to the Galilean motion experiments.

Discovery processes in STERN are instantiated as rules (productions). A scheme based on the framework is used for the definition of the conditions and action of rules in a principled manner. Similarly, tasks comprising groups of rules are defined using the same scheme, with a clear distinction between domain-specific and domain-independent rules. STERN possesses 64 rules, grouped into 16 classes, organized into a task hierarchy.

STERN successfully models the Galilean episodes of discovery. Given the Aristotelian laws as input it finds that they are unacceptable. STERN then goes on to perform experiments to obtain a body of results for generalizing into hypotheses. At this stage STERN makes what is arguably a "genuine" discovery - the law governing the period of swing of pendulums. All the hypotheses obtained by generalization are then analysed in order to discover the correct law of free fall. To show that this law is generally acceptable, STERN has to invent new experiments. The discovery path followed by STERN resembles closely the course that Galileo took.

### 1.6.4 STERN's Discovery Strategies

STERN's discovery abilities can be considered as four main strategies or tasks (subprograms). The most frequently used strategy is the confirmation of existing hypotheses (Chapter 5). This attempts to assess known hypotheses by generating models and instances with respect to particular experimental paradigms and setups. The degree of match between the predictive instances and the experimental tests

forms the basis of the assessment of the acceptability of models, and in turn hypotheses.

STERN obtains new hypotheses using one of two strategies. (i) The data-led approach generalizes experimental results into hypotheses (Chapter 6). Experimental paradigms and setups are selected by STERN and experimental tests are designed. The tests are performed and the results interpreted into instances. The instances are then generalized into hypotheses via models. (ii) The second method is the theory-led generation of new hypotheses from existing hypotheses (Chapter 7). Qualitative and quantitative hypotheses are used and they range over the whole spectrum of acceptability. The free fall law is found using this new hypothesis generation strategy.

STERN tries to confirm the new free fall hypothesis, but is unable to, since the hypothesis is intractable, given the original experimental paradigms that are available. Specifically, a wholly theoretical term in the free fall law cannot be replaced by some expression with only directly measurable terms. STERN chooses to employ the fourth strategy to overcome this problem. The strategy involves the invention of new experimental paradigms (Chapter 8). The new experiments bring together novel ways of combining experimental parameters, which in turn allows STERN to get around the theoretical terms that could not be eliminated.

These four strategies allow STERN to successfully model the Galilean episode. However, it is not simply that STERN possesses them that makes the system so successful; what is particularly important is the way in which the strategies interact over time.

### 1.6.5 The Cognitive Science Of Scientific Discovery

The framework has been used to analyse previous work and it forms the basis of the STERN discovery system. This much shows that it has great utility. Furthermore, it can be used to analyse the nature of the scientific discovery in cognitive science, in a reflexive way (Chapter 9). We will see how all the entities

and types of studies in cognitive psychology and AI map neatly on to the components in the framework. This provides a particular useful way of considering interesting issues about how scientific discovery has so far been studied in cognitive science. We will see in a general way why STERN is a significant advance on previous models. In particular, STERN is a much more complete model of scientific discovery. As such STERN shows that there are likely to be interesting discovery heuristics and abilities that only emerge when systems attain a certain level of complexity.

# Chapter 2

# Previous Work

## 2.1 INTRODUCTION

Within Cognitive Science, scientific discovery is now a well-established and steadily growing area. However, research in the field has been far from homogenous; three distinct approaches can be discerned. First, some researchers have attempted to produce computational models that simulate discovery episodes from the history of science with varying degrees of fidelity (e.g. Langley *et.al.*, 1987; Kulkarni & Simon, 1988). Others have attempted to solve existing scientific problems, using the power of computer technology, with no pretence to model human discovery abilities (e.g. Buchanan & Feigenbaum, 1978). Finally, there are empirical studies of people's discovery abilities carried out in simulated scientific domains where the phenomena are under the control of the experimenter (e.g. Gerwin, 1974; Klahr & Dunbar, 1988). All of this work is relevant in some sense to the present thesis and will be called upon in the body of this chapter. However, the central concern of this thesis is the characterization of scientific discovery in computational systems – *modelling* the processes of discovery. Hence, the first category of research, the computational models, will predominate in the review.

### 2.1.1 Computational Models Of Discovery

Although only a fairly new field in AI, numerous discovery systems have already been developed. The number of programs is not only large (approximately twenty will be considered here) but the variety between them also great. There are many ways we can divide up the body of work formed by these programs. For example, one could group them in terms of the representations they adopt for scientific knowledge; ranging from ECHO's (Thagard, 1989a) propositional

representation to STAHL's (Langley *et.al.*, 1987) predicate-attribute-value triples to the frames of IDS (Nordhausen & Langley, 1987). One could also classify the systems in terms of the architectures they adopt; whether it be production system (BACON, Langley *et.al.*, 1987) or a parallel system (ECHO, Thagard, 1989a).

However, this review will be organized with respect to the processes of scientific discovery that emerge from the proposed framework (see Chapter 1). From this perspective four main concerns can be identified in the literature:

*(§2.2) Data Driven Formation Of Models.* These programs and empirical investigations are concerned with the processes that generate theoretical knowledge, typically laws, from empirical data. In terms of the framework, the processes generalize one or more instances to form a model.

*(§2.3) Theory Driven Model And Instance Generation.* These studies are concerned with processes that use theoretical knowledge often in conjunction with observational data, to generate new theoretical knowledge. Thus, this includes the formation of models from hypotheses and instances.

*(§2.4) Assessing Hypothesis Acceptability.* This is work that models the way the acceptability of hypotheses is assessed in isolation.

*(§2.5) Multiple-Process Models.* Programs in this category all model a rich variety of discovery processes. They typically combine several of the individual processes considered above.

Each of these will be considered in turn later in the chapter. However, before we look at the programs and other research in detail it is worth considering some of the alternative general descriptions of scientific discovery in Cognitive Science that have been proposed.

### 2.1.2 General Accounts Of Scientific Discovery

The high level description of science has not been solely the province of philosophers of science. Some AI research has proposed general characterizations

of scientific discovery.

Langley *et.al.* (1987, 18-20) consider scientific discovery to have four main phases which operate in a cyclical fashion to uncover new scientific knowledge: (i) data is gathered; (ii) parsimonious descriptions of the data are sought; (iii) explanatory theories are formulated; and (iv) these theories are tested. Since testing requires more data the procedure loops back to the first data gathering stage. The middle two phases can sometimes be condensed into a single stage when the parsimonious description acts as an explanatory theory. Furthermore, the data gathering phase comes in two forms: the observation of natural events; and the production of phenomena in experimental apparatus.

Reimann (1990) also considers the process of scientific discovery as a cycle, but one which only consists of three stages. Initially, theories are formulated and predictions are derived from them; then, the data is gathered that bears on the theory; and finally, the theories are tested by comparing the predictions with the data.

The similarity between the cyclical views is plain to see, but this means they also suffer from the same problem. Both characterizations are very general – so general that they do not further our understanding of how scientific discovery occurs. At some vague abstract level the creative process can be said to be cyclical, but to hold that all discovery exactly fits this mould is not a tenable view.

A very different approach to these cyclic views is taken by Holland *et.al.* (1986). They view scientific discovery as a particular form of induction within their Framework for Induction. In this framework, models are represented as condition-action rules and hypotheses are clustered together. Scientific knowledge improves by modifying the strength of rules and hence heightening the likelihood of them being used. It also improves through the generation of rules under special triggering conditions. These conditions, such as the failure of predictions, help to ensure that new rules are relevant. Rules compete or cooperate by a principle of limited

parallelism, and background knowledge plays a role in inference processes. The induction framework is coherent but it is too much like an algorithmic specification of a particular class of programs that are aimed at solving a limited range of problems. It is therefore not particularly revealing about the underlying character of scientific discovery. Further, it is a framework for induction in general, not just scientific discovery, so tends to emphasize some processes that are of minor significance whilst missing out others that may be unique to scientific discovery. These points will become clearer when we consider the PI program later.

These descriptions of science are fairly process-oriented, that is they concentrate on how things happen in discovery. The framework presented in Chapter 1 is different in that it proposes a minimum set of components. The framework leaves open questions such as the specific order of discovery processes to be settled by computational modelling. Some general constraints are placed on the types of processes that can be considered, thus avoiding the vagueness of cyclical descriptions, but not to an extent that rigidly defines the processes even before the investigation has really begun, as in the Induction Framework. The present Framework sails between the two extremes.

But let us now look at the computational models of of scientific discovery in detail, starting with the ones that generalize instances to form models.

## 2.2 DATA DRIVEN FORMATION OF MODELS

A substantial proportion of work in AI on scientific discovery has concentrated on the generalization of instances (often called *data*) into models (typically referred to as *laws*) in situations where no prior theoretical knowledge of the domain exists. In this case the instances are interpreted experimental results, known to philosophers of science as *empirical facts*. Models are consequently parsimonious descriptions that summarize one or more instances. In this section we will see that the programs modelling this task vary greatly in their abilities. We will also see that

in and of themselves they give rather a sparse view of scientific discovery in general. A good proportion of these programs have been developed together under a common approach, which I will call the *BACON* school. It will be considered first.

### 2.2.1 The BACON School

The collection of programs developed by Langley *et.al.* (1987) is called the *BACON* school after BACON, the best known program produced by these investigators. BACON has numerous versions and a suite of sibling programs: GLAUBER, STAHL, DALTON, FAHRENHEIT (Koehn & Zytkow, 1986; Zytkow, 1987) and GELL–MANN (Fisher & Zytkow, forthcoming). Figure 2.1 shows the relationship between the various programs. All were developed under a common AI orientation that has been variously described in Langley *et.al.*(1987, 281-301), Zytkow & Simon (1988) and Langley & Zytkow (1989). It views scientific discovery as a problem solving task to be solved by heuristic search. This is a dramatic shift from their general view of scientific discovery as a cycle of various phases, described above. The need to adopt a different approach for the computation implementation only goes to further emphasize the vacuity of the cyclic view. The problem-solving-by-heuristic-search approach is most suitable for well defined tasks that have states that can be recursively generated and searched by a limited set of rules. The finding of parsimonious descriptions of data is ideal. All the BACON school of programs perform some variation on this basic theme and can be divided into two main groups depending on whether they deal with quantitative or qualitative data. Each class is described in turn.

*2.2.1.1 Quantitative Programs - Five Versions Of BACON Plus FAHRENHEIT*

The BACON program, one of the best known scientific discovery systems, was developed by Langley *et.al.* (1987) and has progressed though six versions. Some are built upon previous versions, but others are substantially independent (see

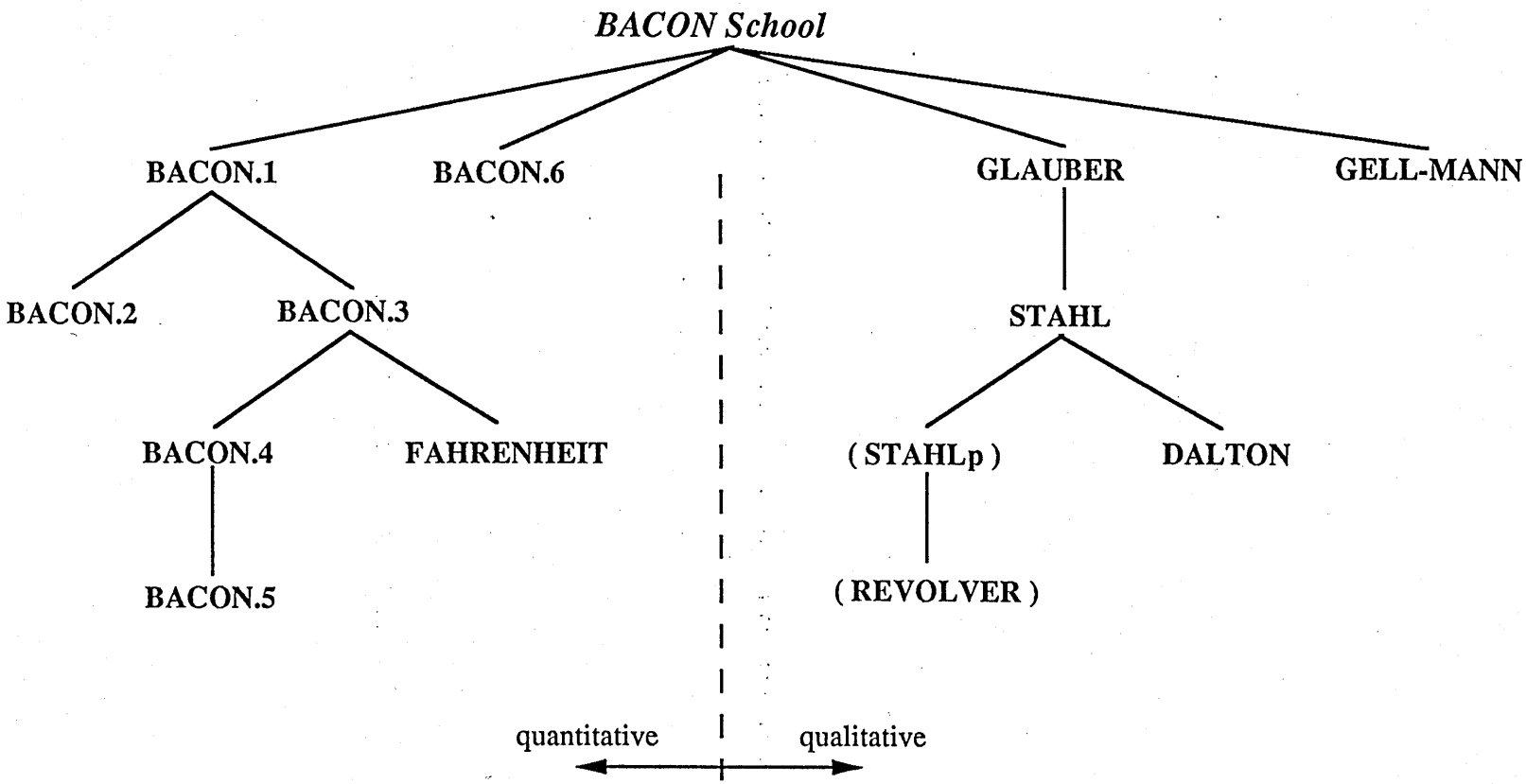



**Figure 2.1 The BACON School & Related Programs**

Programs lower down the hierarchy are developed from higher ones.

Figure 2.1). All versions attempt to find quantitative laws that parsimoniously describe a series of numerical data, the exception being BACON.6 which will be considered later in §2.3.1.1.

The first version of the program, BACON.1, finds a law describing the relation between just two quantitative terms. The sixteen production rules used are grouped into four main categories: data gathering and storage rules; regularity generators; higher-level term generators; and house keeping rules. *Data gathering and storage* mainly involves preparing variables and recording the series of values that have been obtained from the program user. The *regularity generator* heuristics examine the pairs of terms for certain types of regularities: (i) *linearity*, the terms are proportionally related; (ii) *increasing*, the magnitudes of both terms are monotonically increasing; (iii) *decreasing*, the magnitudes of one terms is monotonically increasing whilst the other is decreasing; and (iv) *constancy*, one term is (approximately) constant. Depending on the regularity spotted, the *define new terms and calculate values* group of heuristics forms new terms and calculates their values from the existing terms. The goal is to define a new term that has constant values or that is closer to constancy. For example, when the term $P$ ranges over the values (1, 8, 27) and $D$ ranges over the corresponding series (1, 4, 9), the *increasing* regularity obtains and a new term is defined as the ratio of the terms, that is $D/P$. The values of the new term are then calculated and the regularity spotting process repeated for the $D$ and $D/P$ terms. A law is found when the new term has values that are constant. In the example, when the term $D^3/P^2$ is generated.

Any noise in the data is dealt with during the search for the constancy of a term. This involves a test in which all the calculated values must be within a specified maximum percentage deviation, $\Delta$, of the mean, $M$, of the values. In other words, each value must fall within the interval $[M(1-\Delta), M(1+\Delta)]$ around $M$.

Most of the other versions of BACON are based upon this theme. BACON.2

finds polynomial laws by considering sequential regularities. BACON.3 accepts multiple input terms, rather than just a pair, and so is augmented with a set of heuristics that control the search through the terms in a systematic fashion. BACON.4 has heuristics that can identify intrinsic properties associated with nominal terms (the density of different bodies made of the same material is an example of an intrinsic property). Many scientific laws exhibit symmetry, which BACON.5 exploits to improve its efficiency by assuming symmetry to dramatically cut down the number of combinations of terms it has to search through.

The BACON programs can find quantitative laws from numerical data relating together several terms, perhaps in a symmetrical equation, that may have intrinsic properties and contain some degree of noise. However, their abilities are limited in a number of ways. BACON copes with noisy data by employing the maximum-percentage-deviation technique. The technique has particular deficiencies. For example, a single erroneous value, in otherwise perfect data, may cause the technique to fail if the value falls outside the permitted band. Similarly, high noise levels would require large values of $\Delta$, but this opens up the possibility that incorrect laws might satisfy the constant term test before the correct law is reached. Further problems have been noted by Walker (1987) and Langley *et.al.*(1984) to do with irrelevant terms and the ordering of input terms. In science, discovering which terms are really relevant to a phenomenon is an important part of characterizing a domain. But BACON has no ability to find and eliminate such terms. Rather, it methodically searches though all the terms given as input to the same level of detail at the expense of performance and efficiency. However, even when the input terms are all relevant, the order in which they are presented must be carefully selected by the user, because the intermediate terms in BACON.3 on one level must be linearly related to the input term considered on the next level (Langley *et.al.*, 1987, Chapter 3). Thus the program will fail if the data is presented in the "wrong" order. In effect BACON must be told which terms are independent and

dependent variables. Fortunately, this problem has been tackled in FAHRENHEIT.

FAHRENHEIT (Koehn & Zytkow, 1986; Zytkow, 1987) employs BACON.3 as a subprogram so can finds laws just like BACON. However, FAHRENHEIT also attempts to find the range of values over which the law is valid (in terms of the variables referred to by the law). For example, Black's law of specific heat concerns heat transfer between bodies of different mass, heat capacity and temperature, e.g. mixing mercury and water. However, it is limited to regions where no state transitions occur, such as the boiling of water into steam. For fixed quantities of mercury and water, only the initial temperatures of the substances affect the outcome of the process. So, FAHRENHEIT attempts to determine the region within which Black's law remains valid by finding expressions that describe the boundaries of the law as it just breaks down, when the initial temperatures are varied. An additional feature possessed by FAHRENHEIT is the ability to rearrange the order in which terms are examined. As noted above, the depth first search of BACON can fail to find a law if the user presents input terms in the "wrong" order, but FAHRENHEIT has a second level of search which proposes alternative combinations of terms whenever regularities are not found.

FAHRENHEIT is an advance on BACON as it not only finds a law but attempts to find the region within which it is valid with respect to certain dependent terms. Further, it has the ability to sort out the order of input terms for itself. However, the range of phenomena that can be modelled is very much more restricted than BACON, because the phenomena must have a region bounded by discontinuities. Although FAHRENHEIT considers the range of applicability of a law this should not be confused with the more usual sense in which theoretical knowledge is thought to be true. The description of the range of applicability simply characterizes another aspect of the law. Black's law is typically considered valid because it satisfactorily accounts for many different combinations of substances examined in

different experimental situations. FAHRENHEIT considers just one pair of substances.

*2.2.1.3 Qualitative Programs - GLAUBER, STAHL, DALTON & GELL-MANN*

There are four programs that work on qualitative data in the BACON school. For the sake of brevity just one will be considered in detail as it is representative of the rest. I will consider STAHL (Langley *et.al.*, 1987) (see Figure 2.1) in detail, because it forms the basis of later work on an interesting system called REVOLVER.

STAHL can find models of the chemical compositions of compounds in terms of their elements, given data consisting of ordered sets of reactions. Like GLAUBER and DALTON, STAHL uses a predicate attribute and value notation to represent chemical reactions. For example:

(reacts inputs {charcoal air} outputs {phlogiston ash air}), . . . (2.1)

states the reaction that is assumed to occur during the combustion of charcoal under the pre-oxygen phlogiston theory.

Three main heuristics are employed by STAHL to find models of the compositions of substances. INFER-COMPONENTS is a heuristic that posits that one substance is composed of two others. When a reaction synthesizes the substances B and C into A, or A decomposes into the B and C, INFER-COMPONENTS reasons that A is composed of B and C. Once A is known to be a compound, the SUBSTITUTE heuristic may swap B and C for A in reaction equations. REDUCE removes occurrences of the same substance from both sides of a single reaction equation.

STAHL invokes the three heuristics is a specific order. INFER-COMPONENTS is first applied alone to the input reactions. Then REDUCE followed by INFER-COMPONENTS. Finally, SUBSTITUTE, REDUCE and then INFER-COMPONENTS in that order. The results of each path followed are analysed individually for reaction equations that are consistent; this is to say that their sides balance. Specific backtracking methods are employed to deal with inconsistencies (e.g., nothing on one side of a

reaction equation) and circular definitions (e.g., a compound constituted by itself and another substance). In certain cases identification heuristics may be required to infer, for example, that two substances occurring in two similar equations are in fact the same.

GLAUBER and DALTON also work in the domain of chemistry but operate on different levels. GLAUBER takes specific observations about chemicals (e.g. their tastes and the reactions they have been involved in). It attempts to find classes of chemicals with similar properties and general reactions across those classes. DALTON considers componential models of the structure of substances in terms of constituent molecules and atoms. The descent down the scale of entities continues with the GELL-MANN system (Fisher & Zytkow, forthcoming), that discovers Quark models that explain the properties of families of subatomic particles. Grossly simplified, GELL-MANN proposes models based on different numbers and types of quarks. Those that do not satisfy the *additivity* law (which states that for each attribute, the sum of the values over component quarks must equal the particle's value) are eliminated.

An obvious limitation of the qualitative models in the BACON school is that they only use qualitative reasoning, with the exception of GELL-MANN. It seems highly likely that the power of GLAUBER, STAHL and DALTON could be substantially enhanced if quantitative techniques were to supplement the existing qualitative heuristics. For instance, Proust's law of definite proportions was of great significance in (John) Dalton's thinking and discoveries (Holten & Roller, 1958). Heuristics based on this law may help reduce the DALTON program's search space considerably. The importance of modelling both quantitative and qualitative inferences in a complementary manner within a single system is an issue that recurs several times in this review.

*2.2.1.3 The BACON School Programs As General Models Of Scientific Discovery*

How realistic are the programs in the school as general models of scientific

discovery? Even in terms of the general description of the scientific enterprise given by the BACON school their programs do not cover much ground. The general description considers phases including: (i) data gathering; (ii) finding parsimonious descriptions of the data; (iii) formulating explanatory theories; and (iv) testing these theories. Of the four phases outlined only the second, formation of parsimonious descriptions of data, phase is considered in detail. The data gathering heuristics do not model the observation of natural events or the production of phenomena in experimental apparatus. Furthermore, once a satisfactory law has been found most of the programs do not use the law to make further inferences (the one exception being FAHRENHEIT). Finally, the processes involved in the third and fourth phases, the formulation and testing of explanatory theories are not modelled at all. With respect to the present framework (Chapter 1), the models formulated are not generalized into hypotheses and the acceptability of theoretical knowledge is not assessed.

### 2.2.2. Combined Qualitative & Quantitative Inference

IDS and ABACUS are programs that fit into the data driven formation of models category, but they are more sophisticated than the work of the BACON school. Each program possesses qualitative and quantitative representations and uses qualitative and quantitative heuristics to finds models from instances.

*2.2.2.1 Integrated Discovery System (IDS)*

IDS is an integrated discovery system in that it integrates quantitative and qualitative inferences (Nordhausen and Langley, 1987)[1]. Like FAHRENHEIT, IDS works with phenomena that exhibit discontinuities when some independent variable is increased. For example, IDS attempts to characterize the changes that occur when a quantity of ice is heated, melts into water and eventually boils into steam. The systems begins by characterizing the phenomenon using qualitative

[1]Although Langley helped develop IDS, it does not fit the BACON school mould. The program does not recursively search an homogenous state space using a single group of heuristics.

schemata to record the changes between states as heat is supplied. The schemata, represented as frames, have three slots containing: descriptions of objects present and their phase (e.g. solid ice and liquid water); specifications of the quantitative conditions of attributes of these objects (e.g. weight greater than zero, constant temperature), and process descriptions of changes to variables (e.g. positive rate of weight increase). A new schema is induced whenever a limit point, or state transition, is found. The qualitative schemas are used to infer two types of law: (i) constancies over all objects of a given class, such as the invariant melting point of water across different samples; and (ii) constancies within a single qualitative schema, such as the conservation of mass as ice melts into water.

IDS's quantitative discovery abilities are rather limited. The rich vein of information available in the qualitative schema is barely exploited in the inference to quantitative laws. This limitation is due more to lack of development than to any underlying weakness of the approach. Even so, IDS does begin to demonstrate how the combination of the quantitative and qualitative abilities can lead to more powerful discovery models. For instance, an effective solution to FAHRENHEIT's inability to search for quantitative laws across and between state transitions is now available in IDS.

*2.2.2.2 ABACUS*

ABACUS (Falkenhainer & Michalski, 1986) considers qualitative and quantitative inferences in the reverse order to IDS. The finding of equations that describe numerical data comes first, in a manner that superficially resembles the BACON programs but is more sophisticated in five respects. First, ABACUS can find separate equations for a set of data by searching portions of the data where the relation between the variables is not monotonic. Like BACON regularity-spotting heuristics are used: $Prop^{+}$ and $Prop^{-}$ are analogous to BACON's increase and decrease regularities, and $Prop^{?}$ and *Norel* indicate insufficient data is available or that there is no relation, respectively. Second, the qualitative relations between the

input terms are all found at once, thus eliminating irrelevant terms at an early stage. Third, as an initial best guess, a single equation composed of all the relevant terms is proposed. The equation is based on the noted $Prop^{+}$ and $Prop^{-}$ regularities, with the hope that the main protracted rigorous search may be avoided. Fourth, the main search for equations considers products, ratios, sums and differences of terms (nodes) on a current level and defines new nodes on the next level by proportionality testing of all previous nodes. The search space is reduced by *suspending* those branches that seem farthest from constancy. Fifth, the search space is further constrained by three rules concerned with certain types of equations. (i) Equations that do not have balanced units according to dimensional considerations cannot be valid. They are eliminated. The units for particular terms are user specified inputs. (ii) Some equations are mathematically equivalent, but syntactically different. Only mathematically different equations are retained. (iii) Terms in equations that can be trivially cancelled are cancelled out. ABACUS's equation finding abilities are both extensive and powerful.

The quantification process employs the Aq algorithm (Falkenhainer & Michalski, 1986, 386-8) that has two main stages: the first finds sets of attributes that can distinguish equations from each other; the second takes the best set of attributes for each equation as a positive instance, and finds the maximally general description that does not cover any other equation. The resultant quantification is in effect the precondition for the application of the equation.

The abilities of ABACUS clearly surpass those of BACON. The system copes efficiently with irrelevant variables and does not need to be told which ones are independent or dependent. Multiple equations are found for discontinuous data and quantified in a manner that achieves the same result as FAHRENHEIT's validity expressions. The units of the terms are acknowledged and play a significant role in the identification of valid equations. In fact, this is the first program, of all the ones

so far considered, that really seems to have terms that refer to observed or measured quantities, because of the presence of units. As we will see below, more complete models of discovery need to exploit this technique more fully. The use of units is an example of the correspondence between the theoretical terms in instances and the parameters in experimental tests - the most direct form of communication between the theoretical and experimental components considered by the framework.

We have now seen several computer programs that successfully make generalizations from data, using a wide range of different techniques. But how do humans perform such tasks? Do they use similar methods – like spotting trends in numerical values to find equations? The empirical studies that have studied this task will now be considered. They provide a few answers to these questions.

### 2.2.3 Empirical Studies Of Generalizations From Data

We are considering how instances are generalized in order to form models. Computational systems that perform this task were discussed above. However, this is something that humans scientists are well able to do and have been doing long before the invention of computers. So, it is not surprising that this particular aspect of the human scientific discovery has been investigated empirically. In all the studies, the investigators typically have control over the data presented to the subjects.

#### *2.2.3.1 Equations From Numerical Data*

In the two studies undertaken by Gerwin (1974) and Qin & Simon (1990) subjects were given two lists of related numbers (pairs of cartesian (x,y) coordinates). Gerwin's data was generated from equations like:

$$y = x.\cos x + x^{1/2} + k.e_x , \qquad \ldots (2.2)$$

where $k.e_x$ is a function that adds noise. Qin & Simon's subjects were given data that satisfied Kepler's third law; that is $D^3 / P^2 = C$, where D is the distance from the sun, P the period of revolution and C a constant.

Gerwin's subjects were presented with the data plotted on a graph and suggested

an equation composed of functions from a given list. New data points were calculated from the suggested equation and plotted by the experimenter. Additionally, comparisons between the new and the original data points were plotted. The subject examined the new information and suggested further equations until satisfied that the correct one had been found. The analysis of the protocols over a number of different subjects and data sets permitted Gerwin to write a computer program simulating the behaviour of the subjects. Then both the computer program and new human subjects took part in a second series of similar experimental trials. Both found correct equations with an accuracy of approximately forty per cent, with a close match between the program and human solutions. However, the types and sequence of processes used by the two varied in all but the simplest cases.

Qin & Simon (1990) found that the subjects' behaviour can be described in terms of two levels of search: function spotting and parameter search. Function spotting refers to processes of finding the general form of the equation (e.g., an exponential rather than a sinusoidal function). Diagrams and graphs were typically used for this purpose. Parameter search involves finding the magnitudes of constants.

Three conclusions of relevance to computer modelling can be drawn from the two studies. First, BACON-like regularity-spotting heuristics model human scientists, in certain circumstances, when looking for trends in data and proposing new equations. Second, qualitative techniques (like graphs and diagrams) are used by humans in the interrogation of data, often as a short cut to blindly applying quantitative analysis techniques. Third, the underlying variability in the performance of human subjects is great even in the simple tasks in the experiments. Thus, there is little hope of successfully simulating every little step in processes that bring about a discovery - the discovery paths of Gerwin's program and humans varied substantially. This supports the view that *modelling* is preferable to exact simulation.

*2.2.3.2 WASON's '2 4 6' Experimental Paradigm*

Wason's (1960) '2 4 6' task is the basis of an experimental paradigm that has been used to investigate how scientists propose expressions that explain data. The paradigm has been the subject of an extensive programme of research (see Evans, 1989, for a review). In the basic paradigm the experimenter has a rule in mind that describes the structure of numerical triples (e.g. 2 4 6). The subject attempts to find this rule by proposing triples and being told whether they are instances of the rule. Classically, the rule is 'Any ascending sequence' and the subject is initially told that '2 4 6' is a positive example. Subjects typically posit rules like 'Ascending with equal intervals' and only propose triples that are positive instances of their own rule. This is has been taken as evidence that humans exhibit *confirmatory bias*; namely the fundamental tendency to only seek information consistent with present beliefs. This has been used to argue against the Popperian falsificationist view of scientific discovery (Evans, 1989).

The main criticism to make of research under the Wason paradigm is that the '2 4 6' task only bears a superficial resemblance to real-life scientific discovery. For example, binary feedback about whether the proposed is an instance of the target rules is supposed to represent experimental tests. Experiments provide a rich source of information even when negative outcomes are obtained. Other criticisms along this line are that: the experiment is rigidly predefined; the mapping between experiments and laws is obvious; and there is no parallel of prior knowledge and the semantics of real situations under the Wason paradigm. As we shall see later, Klahr & Dunbar (1988) recognize these limitations and have been motivated to perform a study on a much more realistic scientific discovery context.

We can draw the following general conclusion. Apart from supporting the view that regularity-spotting heuristics may be a reasonable model of certain aspects of instance generalization, the empirical research does not provide much information

that is useful to the computational model builder.

### 2.2.4 Limitations Of Data Generalization Models & Studies

In this section, we have seen a wide range of programs and reviewed empirical studies that have focussed on the inference to models that parsimoniously describe data. However, as general models of scientific discovery they are far from complete. The generalization of data into laws is an important part of scientific discovery but it is by no means all or even the most significant part. Major issues are left unanswered by the research. How are the data obtained in the first place? What happens when the data are unreliable (e.g., noisy)? Typically, these programs take just one set of data as input and find one law, whereas scientists usually gather many sets of data from different experiments and consider multiple models. Furthermore, models themselves also become the subject of inferential processes when their acceptability across different experimental situations is assessed or when they are generalized to form higher-level theoretical knowledge (i.e., instances). Also, laws may be used to help infer new theoretical knowledge.

Fortunately, there are other programs that have considered some of these issues, to which we will now turn.

## 2.3 THEORY DRIVEN MODEL AND INSTANCE GENERATION

The programs examined in this section all possess some theoretical knowledge from the start. The knowledge (hypotheses or models) is used typically in conjunction with data to infer further theoretical knowledge. A mixed bag of programs are considered but they can be divided into groups according to the particular task to which they put their theoretical knowledge. The first group deals with the modification of unacceptable models into forms that more closely account for phenomena. The second group employs valid hypotheses that are used to generate models, or valid models that are used to generate instances, relating to a specific situation.

### 2.3.1 Modification Of Given Models

The programs here all possess a model that is almost adequate and use data in order to modify and improve that model.

*2.3.1.1 BACON.6*

This version of BACON (Langley *et.al.*, 1986) differs from the previous versions (Langley *et.al.*, 1987) because it does not find a model from data but is given the correct model as its input; for example $Y = aX + b$. However, the values of the constants *a* and *b* are unknown so BACON.6 attempts to find the values that give the best fit to the data. Thus, the program searches through a space of possible combinations of values of the constants, where each state in the space differs from a previously visited state with respect to the value of just one constant. New constant values are calculated by incrementing (or decrementing) the value by a geometrically decreasing amount. The control strategy here is a beam search version of hill climbing, with termination when an invariant set of constant values is found.

The task that this version of BACON performs is limited, but even the manner in which it is performed is very inefficient. Scientists are likely to have some idea of which constant is most significant and therefore consider a smaller number of combinations. Important constants will likely emerge naturally when a wider context of scientific discovery is modelled.

*2.3.1.2 STAHLp & REVOLVER*

Unlike the other qualitative programs of the BACON school, STAHL's abilities have been developed further. The program becomes STAHLp in Rose & Langley (1986) and is improved by augmenting the representation of reactions with *reduced lists* and *source tags*. The reduced lists store details about which substances are eliminated from reactions or componential models by the REDUCE heuristic. Source tags indicate at what stage each substance was first considered. This information improves the accuracy of the program in its actual generation of models and permits

the use of an enhanced mechanism for recovering from erroneous inferences.

In Rose (1988a) STAHLp evolves into the REVOLVER system. The new program uses an evaluation function to assess which of the alternative models should be modified. The evaluation function calculates a value for each model based on factors such as the number of beliefs supported and the number of substances in the reaction. This value measures the desirability of modifying each model using criteria concerned with the minimum mutilation of the data base, conservatism and complexity. Two further REVOLVER enhancements (Rose, 1988b) have been implemented. First, the program can cope with multiple models considered by different agents using degrees of belief supplied as inputs. The fixed belief values help to order the sequence in which the models are considered. Second, REVOLVER is given the ability to postulate new substances when certain types of inconsistency arise during inferencing.

In conclusion, REVOLVER possesses sophisticated theory revision abilities but they are mostly domain-specific. The latest version has the user input degrees of belief assigned to models that are somewhat like measures of acceptability. However, the degrees of belief are not amended during the model revision or any other process. In addition to REVOLVER, only ECHO (Thagard, 1989a: see below) has attempted to model separate agents in scientific discovery. However, this ability in REVOLVER is limited and can be viewed merely as a mechanism for partitioning and storing sets of premises that are dealt with sequentially. An adequate model of separate researchers working on the phenomenon would require intercommunication between the agents; with, for example, specific challenges and defenses of particular premises.

*2.3.1.3 COPER*

COPER (Kokar, 1986) takes an equation as input and uses dimensional considerations to determine whether it has missing or redundant arguments. Briefly,

dimensional analysis is based on the fact that equations normally relate together terms standing for independent physical quantities (e.g., length, mass, time). The fundamental point to note is that the units on both sides of the equation must balance. COPER's abilities to find missing or redundant terms relies on tests for dimensional consistency in equations.

COPER can make various inferences about equations just by examining their dimensionality. The program can do this because of its knowledge of the units of theoretical terms. This knowledge is derived from the fact that experimental quantities are defined on scales of measurements using base units (e.g., the metre, kilogram, second). As such, Coper is another case which indicates the importance of theoretical inferences having access to knowledge about experiments through the correspondence between theoretical terms and experimental parameters.

### 2.3.2 Generating Models & Instances From Hypotheses & Models

We have seen how theoretical knowledge can be found by the generalization of instances to form models. Now we will look at another way in which theoretical knowledge can be generated. In particular, we will consider how hypotheses or models can be used to generate less general models or instances. In terms of the present framework all the programs perform similar tasks, so for the sake of exposition the approach adopted by particular groups of research will be used to classify the systems.

#### *2.3.2.1 Engineering Systems*

Many AI programs have been development that solve real scientific problems using the sheer information processing power of computers. The researchers who adopt this approach have no intention of modelling human capabilities, so their programs can be called *Engineering Systems*. Engineering systems employ established scientific knowledge to make specific discoveries in well delimited domains.

Just one such discovery system, MetaDendral, is considered in detail in this

section. Other examples of engineering AI discovery programs are: PROSPECTOR (Duda *et.al.*, 1979; Campbell *et.al.*, 1982), which was designed to help geologists in mineral exploration and has successfully predicted ore deposits; MOLGEN (Friedland & Kedes, 1985), which acts as an intelligent assistant for molecular biologists; and PROTEAN (Hayes-Roth *et.al.*, 1986), which derives protein structure from constraints.

MetaDendral's (Buchanan & Feigenbaum, 1978) task is to find rules for the fragmentation of molecules in mass spectrometry. The inputs are structural descriptions of molecules and their mass spectra. The program is composed of three sub-programs that are called in sequence to perform particular tasks. The first sub-program simulates a run of a molecule through a mass spectrometer to find the points at which the molecules are cleaved. The second sub-program generates rules that describe the bond environment surrounding each break, starting with the most general rule that can be considered as an hypothesis. The final sub-program refines the rules by generalization or specialization in comparison with real mass spectrographs, and ranks them according to their predictive accuracy. MetaDendral has successfully found previously unknown rules for certain classes of molecules, and has been modified to perform Carbon-13 analysis of molecular structure in organic chemistry (Gray, 1984).

Gray *et.al.* (1988) consider MetaDendral as an example of Explanation-Based Generalization (e.g. Mitchell *et.al.*, 1986). EBG is a technique that attempts to explain data (mass spectra) by forming semi-empirical laws (the fragmentation rules) from a domain theory (the hypothesis). The technique is established in AI but whether it is applicable to the general modelling of theory-led discovery is debatable. It is seldom the case in scientific discovery that acceptable hypotheses and instances are both available when no intermediate models exist. When an hypothesis precedes other things scientists tend to generate predictive models before data is gathered. Similarly, when only instances exist they tend to be generalized to

form models before hypotheses are inferred (however see Rajomoney *et.al.*, 1985).

Scientific discovery has not been the only form of discovery that has been considered in AI. Two important discovery programs need to be mentioned that work in domains that are not strictly scientific. They are Lenat 's AM and EURISKO (Davis & Lenat, 1982; Lenat 1983; Lenat & Brown, 1984). Both are given as input some detailed information about their discovery domains, and Lenat does not intend them to be considered as direct models of human discovery abilities, so his programs are *engineering systems*. AM and EURISKO are particularly effective and successful designs for discovery systems. They use concepts represented as frames, manipulated by heuristics expressed as productions, with a best-first search of tasks guided by heuristics that modify "interestingness" and "worth" of concept slots. AM makes discoveries in pure mathematics and EURISKO has, for example, successfully played a war game that involves designing battle fleets.

*2.3.2.2 Models Of Problem Solving*

There has been much research in Cognitive Science on scientific problem solving which typically takes a theory-led approach in which high-level principles, such as Newton's laws, are applied to particular situations.

Larkin *et.al.* (1980) model the behaviour of expert and novice humans in the domain of kinematic and dynamic physics problems. Their program, implemented as a production system, starts with a problem and a stock of principles. (For example, the problem might state that a block is sliding down a ramp with a certain coefficient of friction and that the speed after a certain period of time is desired.) The set of relevant principles are applied successively until all the unknown variables can be eliminated (the program does not actually manipulate the equations). The difference between novices and experts is captured by two

strategies. Novice behaviour is characterized by *means-ends analysis* in which principles are selected according whether they reduce the difference between the current problem state and the desired goal. The experts are modelled by a *knowledge development strategy* that selects principles which permit the finding of the value of a new variable. In this way new information is generated at each stage that follows directly from the known equations. (See Bundy *et.al.*, 1979, and Luger, 1980, for other programs in this domain.)

Jones & Langley (1988) take a different approach and attempt to build certain properties of human problem solving behaviour into their EUREKA program. Four properties are considered: the use of heuristic methods; being non-systematic; performance improvement with experience; and, being sufficiently insightful to respond to external stimuli. Specific techniques are used to model each property. For example, EUREKA models the non-systematic property by not backtracking whilst problem solving using means-ends analysis. Although an interesting approach it does rather beg the questions being investigated in scientific problem solving.

Clearly, scientific problem solving and scientific discovery are related. In the context of theory-led inferences, finding equations to solve a particular problem is akin to the generating of predictions from hypotheses in order to account for a particular experimental paradigm. Nevertheless, they differ in an important respect. Problem solvers assume the stock of principles to be true when they infer an equation to describe the motion of a block sliding down a ramp. However, in discovery the aim is to determine which of the principles, if any, are acceptable. Which strategies scientists performing discovery follow - means-end analysis or knowledge development - and whether the uncertainty about the truth of hypotheses influences the strategies, are open questions for the future.

Another area of AI that treats problem solving is Qualitative Reasoning. As the name implies the approach involves purely qualitative attempts to build models of

physical devices (such as regulators) and to perform various kinds of reasoning task on the models. (See Cohn, 1989, for a review.)

The modelling of scientific problem solving is closely related to scientific discovery but they are different. Problem solving tends to start with accepted true theoretical knowledge, whereas such knowledge must first be uncovered in discovery. The application of this knowledge to particular situations in problem solving aims to explain or account for the situation, whereas in discovery the aim is also to test the validity of the knowledge.

*2.3.2.3 An Architecture For Theory Driven Scientific Discovery*

An alternative to the above problem solving approach is Sleeman *et.al.*'s (1989) architecture for theory-driven scientific discovery based on the analysis of an episode of discovery. BLAGDEN is a system being developed with this architecture. Discovery starts with a weak theory (e.g. a Newtonian description of planetary motion in the solar system) that is an instance of an acceptable core theory (Newton's laws). The weak theory needs to be adapted to account for a new situation (the discovery of a new planet) by proposing *informal qualitative models*. These models identify the relevant dependent terms (such as periods of revolution) and help to specify *law frameworks*. A law framework delimits the space of quantitative laws. Finally, input data is used to infer the actual law. Throughout the procedure the core theory and background knowledge are used in the various inference steps.

The fact that the architecture is based on the analysis of a particular episode of discovery means it is likely to be a successful model. However, it also suggests that it is unlikely to be more generally applicable to other cases of discovery. Further, as in the problem solving models, the core theory is given as input and assumed to be correct, whereas an important part of theory-driven discovery is the demonstration that high-level theoretical knowledge is acceptable in the first place.

*2.3.2.4 Limited Experimental Design*

To finish the review of programs in this section, we will briefly consider a theory-led system that differs from the rest, because it has a limited ability to design experiments. Rajamoney *et.al.*'s (1985) system (with no name) has beliefs about processes involving fluids, such as flow and evaporation. The system makes predictions using the beliefs that apply to specific situations, in order to test their validity. When a prediction fails the system attempts to find out why, by examining the beliefs that are closest to the unexpected behaviour. This is done by designing experiments to differentiate between the processes. For example, by maximizing the surface area in contact with the air, whilst minimizing the contact area with the vessel, the system can distinguish between evaporation and absorption. This is certainly an interesting approach, but unfortunately it has not been explored in any depth, or applied to other domains.

### 2.3.3 Limitations Of The Theory-Led Discovery Systems

The obvious criticism of the theory-led discovery systems is that they do not model how the theoretical knowledge, supplied as input, is generated in the first place. More productively, we can say that an adequate model of scientific discovery must include processes for both data-led inference to theoretical knowledge (as considered in §2.2) as well as theory-led processes that apply such knowledge.

Even though we have seen, in this section, many different types of inferences using theory it is clear that some aspects are absent. First, the assessment of the acceptability of theoretical knowledge has not been modelled; the programs usually proceed directly to a single answer that is assumed to be correct. Second, the role of experiments is absent, their only manifestation takes the form of true and accurate experimental results. Nor does theory-led discovery model the generation of new theories from unacceptable theories. BACON.6, REVOLVER and COPER improve given theories but no programs use the information about a phenomenon that is

encapsulated in partly or totally unacceptable theories. Unacceptable theories may indicate that portions of the hypothesis space need not be searched as the terms to which they refer could be irrelevant. Similarly, partially acceptable theories may suggest which terms are likely to be relevant – this is an idea underlying Sleeman's theory-driven discovery architecture.

Fortunately, the assessment of acceptability of theories has been considered in Cognitive Science.

## 2.4 ASSESSING HYPOTHESIS ACCEPTABILITY & ECHO

We will now consider the assessment of the acceptability of theories by a system that does not also model how the theories were obtained in the first place.

Thagard (1989a) proposes a Theory of Explanatory Coherence that is implemented in the ECHO computer program. The program assesses the acceptability of competing mature research programmes that have investigated the same set of phenomena; for example when the oxygen theory of combustion was becoming a real challenge to the previously dominant Phlogiston theory.

The theory, stated as seven principles, considers the coherence of explanations within a scientific theory as the basis for judging the acceptability of theories. Explanatory coherence relations between propositions are symmetrical and several propositions are deemed to cohere if they explain a common proposition. Propositions *incohere* when propositions are contradictory. Observations or data are assumed to have their own acceptability (a data priority principle) and there is coherence associated with analogies. The specific acceptability of a proposition is determined by its coherence with the system it is in and the overall coherence of a system is a function of the pairwise coherence of its propositions.

To maintain the holistic nature of the theory, Thagard adopts the connectionist paradigm to implement the theory in ECHO. Like all connectionist systems, ECHO possesses nodes and links; the nodes represent propositions and links the

explanatory coherence relations. The weight or activation of links is positive for coherence and negative for contradiction. The nodes representing observational data are connected to special nodes that have a constant level of activation to instantiate the data priority principle. Analogies are modelled in ECHO by providing links between pairs of propositions that are similar, with a user set 'analogy impact' parameter that adjusts the significance of these groups. Each run of the network begins with an even distribution of the activation of nodes and a standard connectionist algorithm is used to update the activations at each cycle. ECHO halts, typically after many tens of cycles, when the activation levels of all the nodes have reached asymptotes. The degree of activation of the nodes indicates the relative acceptability of the propositions.

Normally ECHO is used to model the competing acceptability of the two sides of a scientific debate. The user analyses historical material, seeking the main concepts, data, arguments and explanations involved, and then encodes this in a network. The range of episodes is extensive, including: the oxygen phlogiston debate (Thagard, 1989a & in press); Darwin versus creationism (Thagard, 1989a); dinosaur extinction (Thagard, 1988b & 1989a); the continental drift debate (Thagard, forthcoming); and, two examples of trial jury reasoning (Thagard, 1989a).

A host of criticisms of the theory of explanatory coherence and ECHO can be found in the open peer commentaries that accompany Thagard's (1989a) *Behavioral and Brain Science* article (see e.g., Cheng & Keane, 1989b). Here, three criticisms not previously considered will focus on the adequacy of ECHO as a *general* model of the assessment of the acceptability of theories.

First, a generally adequate theory of the assessment of acceptability should be applicable to all stages in scientific discovery, not just the revolutionary periods considered by Thagard. Consider, for example, the assessment of a theory as new evidence and explanations are incorporated over time. Should ECHO (i) be run from scratch each time a new node or link is required, or (ii) can new elements be

added during a particular run when appropriate? Both options are problematic. The first may lead to the abandonment of a theory in its infancy even though it may in the long run be the most acceptable (e.g. the early development of oxygen theory in the presence of the phlogiston precursor). To prevent such occurrences, explanatory coherence will require something like a disbelief suspension mechanism, but this goes well beyond the scope and principles of the original theory. The second option (apparently favoured by Thagard; see Ranney & Thagard, 1988) leads to the problem of interpreting how the cycles of updating activation represent time or map onto events in a discovery episode. This will inevitably be arbitrary as the theory is atemporal. Thus, it can be concluded that ECHO cannot adequately model acceptability assessment in general without *ad hoc* assumptions. For this, and other reasons, Cheng & Keane (1989b) contend that a symbolic approach may be more adequate.

Second, although Thagard claims that the successful modelling of many episodes by ECHO demonstrates the validity of the Theory of Explanatory Coherence, it is by no means conclusive. The theory claims that it is not only explanatory breadth that measures acceptability, but that simplicity, analogies and contradictions have a substantial role. These properties should have a substantive role in the selection of the most acceptable theory by ECHO. However, this is not the case. Table 2.1 shows the numbers of data propositions explained by competing theories in episodes of discovery modelled by ECHO. The theories that ECHO finds most acceptable always possess the greatest number of data propositions. So explanatory breadth alone is sufficient to distinguish acceptable theories; ECHO could achieve the same result just by counting the numbers of data propositions. Furthermore, Thagard has analysed the relative contributions of the various aspects of Explanatory Coherence (Thagard, forthcoming, Table 10.2) and finds that explanatory breadth is by far the most important, with simplicity occasionally

**Table 2.1 Evidence Proposition Numbers In ECHO**

| Episode (reference) | Participants* | Evidence Propositions Explained |
|---|---|---|
| Oxygen-Phlogiston | Lavoisier - oxygen | 8 |
| (Thagard, 1989a) | Phlogiston theorists | 3 |
| Evolution | Darwin | 13 |
| (Thagard, 1989a) | Creationists | 4 |
| Dinosaur debate | Comet | 9 |
| (Thagard, 1988) | Volcano | 3 |
| Dinosaurs revisited | Terrestrial | 13 |
| (Thagard, 1989b) | Comet | 7 |
| Continental Drift | Wegner | 20 |
| (Thagard, forthcoming) | Fixists | 10 |

*ECHO finds the top participant in each episode the most acceptable.

having a role, and the rest being of minor or no utility. This damaging criticism could be parried by ECHO successfully modelling a real episode of discovery where one theory is clearly more acceptable than another, even though they both explain the same amount of evidence.Whether such a case exists and how ECHO would deal with it remains to be seen.

Third, Explanatory Coherence assumes that empirical data have an inherent acceptability. However, this view is not justified, because the strength of support given to a theory accounting for an experimental result varies according to the reliability of the experiment. Scientists are wary of experimental phenomena that have not been reliably demonstrated and are able to judge the degree to which experimental evidence is acceptable, which is by no means a constant across all types of experiment.

In terms of the present framework ECHO only considers the theoretical side of scientific discovery. The data propositions are instances, and all other types of proposition are either models or hypotheses. The problem with mapping Thagard's representations onto the framework is that his propositions do not distinguish any levels of theoretical knowledge.

In summary, Thagard has attempted to develop a theory that can account for how scientists judge the acceptability of theories. However, it suffers from just being an account of acceptability: the incremental development of theories presents a real problem; it is not clear that ECHO requires anything beyond the principle of explanatory breadth to explain its abilities; and the role of knowledge in the reliability of experimental data is glossed over.

In the next section we will consider systems that are much more complete. Not only do they assess the acceptability of theories, but they also discover that knowledge in the first place, and begin to consider the representation of experiments.

## 2.5 MULTIPLE-PROCESS MODELS

The most complete models of scientific discovery so far developed are now considered. The programs have abilities that include: the generalization of instances to form models; the generation of models and instances from higher level theoretical knowledge; and the assessing of the acceptability of theories. The four models to be considered are PI (Thagard, 1988a), HDD (Reimann, 1990), SDDS (Klahr & Dunbar, 1988) and KEKEDA (Kulkarni & Simon, 1990). The first is an example of the Induction framework of Holland *et.al.* (1986) described above in §2.1.3; the next two were based on empirical studies of subjects performing simulated discovery tasks; and the fourth is a detailed simulation of a well- documented episode of discovery.

### 2.5.1 PI

PI (Thagard & Holyoak, 1985; Thagard, 1988a) stands for Process of Induction, which is apt as it is an example of Holland *et.al.*'s, (1986) Induction framework. The program has been used to model the formulation of primitive scientific concepts, such as the wave theory of sound.

Three types of scientific knowledge are posited that have frame-like representations: messages, laws, and concepts. Messages hold the results of observation and inferences. Laws are represented as if-then statements. In English, an example of a law is, if x is copper then x conducts electricity. Concept frames include information about superordinate and subordinate concepts. The frames of all three types have multiple slots; one of the most important of them is a slot indicating the frame's level of activation. When this level is above a threshold the frame comes under direct scrutiny.

PI runs in a cyclical manner with sets of processes repeated at each time step. During each cycle, PI matches the active messages produced by rules fired in the last cycle (or stated in the problem) with all the conditions of rules stored in active concept frames. The rules that can be fired are fired, according to PI's limited form

of parallelism, to produce new messages. An automatic mechanism spreads activation throughout the network to related problems, concepts, laws and messages. For example, concepts with rules that have been fired have their activation levels increased. This activation spreading draws in potentially useful concepts and rules from up and down the conceptual hierarchy and initiates new sub-goals. These new problems may be analogous to previously solved problems so are reactivated to help with the current task. PI monitors the currently active items and may trigger various forms of induction including: instance and condition-based generalization, abduction, and conceptual combination.

The acceptability of laws is assessed using three criteria. (i) The explanatory breadth or consilience of a law (L) (the number of facts explained by the law). (ii) Simplicity, given by:

$$\text{simplicity} = \frac{\text{No. facts explained by L - No. co-siblings of L}}{\text{No. facts explained by L}} \quad \ldots (2.6)$$

L's co-siblings on the same level (ie. co-"hypotheses") are subtracted, because they are likely to be special assumptions accounting for single facts, and therefore detract from the explanatory range of L. (iii) The overall explanatory power of a law is given by the product of explanatory breadth and simplicity. These three measures are calculated for laws on each cycle and are used in the selection of rules to be fired.

*2.5.1.2 Limitations Of PI*

The main criticisms of PI arise mainly from the fact that it is a model based on the Induction framework. The suitability of one of its fundamental tenets for modelling scientific discovery will be questioned. Consider the representation of knowledge in the form of condition-action rules. Although rules are a general way to represent knowledge, it is questionable whether they are the most appropriate

form of representation for all kinds of scientific knowledge. Quantitative knowledge in the form of equations is a particular problem. For example, consider Newton's second law, often stated as F=ma. As a rule it is: if *F* is a force and *m* is a mass and *a* is an acceleration and the magnitude of m and a are known, then the magnitude of *F* is the product of *m* and *a*. However, *F* may be known and either *m* or *a* unknown, so a total of three rules is required to cope with all the combinations of terms (or one rule with very unwieldy disjunctive tests in both its condition and action). It is thus far more economical to express the law in a realistic and directly manipulable form. This allows general rules modelling scientist's mathematical abilities to be employed, which rearrange and substitute values into the equations as required. To summarize, rules are not a straightforward form of representation and they require extra interpretation not needed in more natural forms of expression.

The next program to be considered also uses rules to represent laws and is limited for the same reason.

### 2.5.2 HDD

#### *2.5.2.1 Empirical Study Of An Optics Experiment*

The basis of the Hypothesis Driven Discovery (HDD) model (Reimann, 1990) is the empirical findings of a study performed using a simulated, optical experiment environment. (Reimann's sense of *hypothesis* is like *model* in the current framework). The environment created in the program called REFRACT permits subjects to investigate the refraction of light rays travelling into different media (e.g. glass, diamond) with different shapes (e.g. plane, concaved and convexed). The underlying behaviour is given by Snell's law, but simplified by ignoring the sines of terms (ie. incidence angle / refraction angle = a constant). Qualitative or quantitative predictions can be made and are graphically compared to the actual result. It was found that qualitative feed-back from trials is important for all subjects. Successful subjects differed from unsuccessful subjects in that they: (i)

tended only to vary one independent variable at a time; (ii) they paid more attention to the current hypothesis during experiment design; (iii) they preferred numerical rather than graphical or qualitative predictions; (iv) they were more willing to make generalizations over several experimental results; and, (vi) they found a more complete set of relations between terms. These results were used to guide the construction of a discovery model.

*2.5.2.2 The Computer Model*

The HDD model is a production system with three main conceptual components. First, the *run experiment* component designs an experiment, makes a prediction from a chosen hypothesis, and compares the prediction and experimental outcome. Second, the *evaluate and modify hypothesis* component assesses the acceptability of the hypothesis and attempts to improve it if the prediction failed. Third, the *hypothesis generation* component infers new hypotheses from previous experimental outcomes. These three components operate in a cycle.

The program possesses a representation for experiments but does not have heuristics for the design or performance of experiments. An experiment is defined by the specification of: the optical medium; the values of variables; and which variables are independent and dependent. The user supplies different designs on each cycle and the program makes a quantitative prediction (as HDD has no qualitative reasoning abilities). The experimental outcome, supplied by the user, is compared with the prediction.

Before considering how hypotheses are evaluated and modified let us consider the condition-action rule representation of hypotheses in HDD. The condition specifies the attributes (medium, angles, distances) considered by an hypothesis, which are assigned symbols standing for particular variables or specific values. The action part is an equation of the form:

$$\text{Variable}_1 = \text{Variable}_2 \text{ (op) constant}, \qquad \ldots (2.3)$$

where the *variables* may be distances or angles, the *constant* is a real number, and

*(op)* is an arithmetic operator (i.e., *, /, + or -). This is the only form of equation HDD knows.

Evaluation takes the form of increasing the strength of hypotheses when the prediction is successful and decreasing the strength when it is not. The strength is used to select hypotheses for consideration or to eliminate unacceptable ones when their strength gets too low. The hypothesis is itself modified when the prediction fails by the specialization of the condition part of the rule.

The generation of new hypotheses employs BACON-like regularity spotters that compare the values of pairs of terms from different experiments. *Trend.Direct1* and *Trend.Inverse1* are analogous to BACON's *increase* and *decrease* heuristics (see §2.2.1.1 above). Equations of the form given by equation (2.3) are generated for the appropriate trend. HDD constrains the space of hypotheses by preferring equations that are products or quotients and ones that have the same type of term on both sides of the equation (e.g. both distances).

The program successfully models the performance of the prototypical subjects in the REFRACT experiments, but it has some problems and limitations

*2.5.2.3 Problems & Limitations*

REFRACT is a good simulated scientific discovery environment. In particular, the combination of graphical qualitative information and quantitative data means that subjects could investigate and reason about the phenomenon in more "natural" ways than previous studies such as Wason's (1960) paradigm. However, only quantitative representations and inference were implemented in the HDD model. As we have already seen above, the combination of both qualitative and quantitative inferences is desirable in, if not essential for, powerful and efficient discovery systems.

HDD's representation of experiment is an advance on the programs mentioned earlier, but as the system possesses no abilities to design, perform or even select

experiments, it is only a minor improvement. Furthermore, the range of equations considered by the program is very limited; only two variables related by simple arithmetic operators are considered. However, the program possesses a rich representation of types of theoretical knowledge. For example, every variable has *role*, *type* and *status* properties in addition to its particular value (Reimann, 1990, 87).

The pattern of discovery found in the empirical study and modelled in HDD is a neat sequence of processes that remains constant in every cycle. However, this cycle emerges from the structure of the empirical study. The fact that a prediction must be made before experimental feedback is received forces the subject into the cycle. Scientists do not normally face such a restriction as they may perform many experiments without making any predictions in order to explore the space of experimental results independently of a theory.

The next model to be considered is also based on an empirical study but is more flexible in permitting various different discovery paths to be followed.

### 2.5.3 SDDS

The Scientific Discovery as Dual space Search (Klahr & Dunbar, 1988) model is the result of an empirical study that attempts to overcome the limitations of Wason's '2 4 6' experimental paradigm (see §2.2.3.2 above). (Klahr & Dunbar's reference to hypotheses has been replaced by 'proposition' in this subsection to avoid confusion with the term as used in the scientific discovery framework.)

#### *2.5.3.1 Robot Based Empirical Study*

Klahr & Dunbar (1988) performed a study on a task that is a better representation of a real scientific context. In two consecutive studies, human subjects investigated the behaviour of a computer-controlled robot, attempting to discover the function of a particular instruction in a LOGO-like language. The subject writes a series of instructions using simple commands including the "mystery" function, and observes the consequent movements of the robot. Propositions describing the

function are formulated and subjects are allowed to carry out repeated tests until the correct operational description of the function is found.

Detailed analyses of subjects' protocols led Klahr & Dunbar (1988) to identify two distinct groups of subjects. One group of subjects were *theorists*; they had a "theory-led" approach which involved proposing new propositions and testing them. Klahr & Dunbar called this a proposition-space search. The other group of subjects were *experimenters* and tended to be "data-driven": performing experiments and attempting to infer propositions from the results. Theoretically, they were searching in an experiment-space. Overall, they propose that scientific reasoning can be characterized as dual space search of the physical possibilities of the experimental situation and the space of conceivable propositions.

This dual-space proposal leads to two predictions: (i) it is possible to think of the correct proposition just by a proposition-space search, without using any experimental results, given the overall context of the experimental situation; (ii) when proposition-space search search fails, subjects will switch to experiment-space search. The predictions were tested by Klahr & Dunbar in a second series of experiments and found to have support.

*2.5.3.2 The Model And Processes Hierarchy*

In a similar manner to Reimann (1990), Klahr & Dunbar (1988) have produced a model based on their psychological findings, although they have not implemented it in a running program. They have also formulated a representation that helps to explain the processes involved. Propositions are considered as frames with four slots relating to particular attributes of the mystery function. Klahr & Dunbar have analysed the types of inferences made by subjects in terms of transformations of these frames.

SDDS characterizes Scientific Discovery as a Dual Search of the proposition and the experiment spaces. This search comprises of three main components called, *space proposition search*, *test proposition*, and *evaluate evidence*. The search of

## Table 2.2 SDDS Processes

| Process Name† | SDDS Process Description | Framework Interpretation |
|---|---|---|
| SEARCH HYPOTHESIS SPACE* | Full specification of a proposition using two processes to either generate a new frame or change slot values | Make a complete model from scratch or modify an existing a model. |
| • GENERATE FRAME* | Generation of a new frame using one of two sub-processes. | Make a new model. |
| • EVOKE FRAME | Memory search for information to permit new frame to be constructed. | Search stored models for suitable base. |
| • INDUCE FRAME | Generates a new frame by induction out of a series of outcomes, using two subprocesses in turn. | Make new model from instance(s). |
| ° GENERATE OUTCOME | A combination of processes, see below, that yields input to next processes. | Obtain instance(s). |
| ° GENERALIZE OUTCOME | Generalise over the outcomes in an attempt to produce a new frame. | Generalise to model from instances. |
| • ASSIGN SLOT VALUES* | For a partially instantiated frame, using one of two sub-processes required to make a fully specified frame. | Fully specify partial model. |
| • USE PRIOR KNOWLEDGE | Assign slot values using prior knowledge. | Use Background knowledge to specify model. |
| • USE EXP OUTCOMES* | Assign slot values using previous or new specific experimental outcomes. | Use existing or new instances to specify model. |
| ° USE OLD OUTCOMES | Examine old experimental outcomes to determine specific slot values. | Try old instances to specify model |
| ° GENERATE OUTCOME | See below | |
| TEST PROPOSITION | A series of three processes to formulate an experiment, make a prediction and runs the experiment. | Test model by generating instances and comparing them with expt. tests. |
| • ESPACE MOVE* | See below. | |
| • MAKE PREDICTION | Take current proposition and experiment to make a prediction centred on the focal values. | Generate instance from model and specified expt. setup. |

continued . . .

## Table 2.2 SDDS Processes Continued

| Process Name† | SDDS Process Description | Framework Interpretation |
|---|---|---|
| • RUN | }See below | |
| • OBSERVE | }See below | |
| • MATCH | Note discrepancies between predicted and observed behaviours. | Compares instance and experimental test. |
| EVALUATE EVIDENCE | Determines whether the cumulative experimental evidence is sufficient for acceptance or rejection of the current proposition. | Are there enough instances supporting the present model? |
| • REVIEW OUTCOMES | Consider previous experimental outcomes. | Look at previous instances |
| • DECIDE | Choose whether to accept, reject proposition or continue testing. | Continue, accept, or reject model on adequacy terms. |
| **DEEPER NESTED PROCESSES¶** | | |
| GENERATE OUTCOME | Generates an experimental outcome using three processes. | Obtain experimental test or instance. |
| • ESPACE MOVE* | See below. | |
| RUN | Performs the experiment. | Obtain an expt. test output parameter values. |
| OBSERVE | Note observed behaviours. | Note relation between input-m and output of expt. test. |
| ESPACE MOVE | Two processes for designing experiments. | Full specification of expt. setup. |
| • FOCUS | Concentrates on the most "important" slot of the current frame. | Choose input-m and output parameters. |
| • CHOOSE & SET | Chooses an "important" slot & sets its value, and fixes the rest of the slots. | Specify fixed input-c parameters. |

**Notes and Key**

Adapted from Klahr and Dunbar (1988)

† - Indentation of name indicates the level of nesting in SDDS.

¶ - Indentation indicates depth relative to the first process in section.

• - 1st level of nesting  • - 2nd level of nesting  ° - 3rd level of nesting

* Process includes a conditional test for which subprocess to execute.

the proposition space involves the generation of new frames or the modification of existing ones. To test a proposition an experiment is designed, a prediction made, the experiment performed, and the prediction and observation compared. The evaluation of evidence reviews the present and previous outcomes of experiments and decides if an adequate description of the mystery function has been found. Fuller descriptions of these processes, and their sub-processes, are given in the second column of Table 2.2 (the third column is referred to later in the thesis). Furthermore, Figure 2.2 shows all the processes arranged in the hierarchy proposed by Klahr & Dunbar, with the groups of processes that are repeated at more than one location deliberately highlighted. One set relates to the design of experiments (ESPACE MOVE). It occurs within both the *proposition space search* and the *test proposition* branches of the hierarchy. This clearly demonstrates that theory and experiment interact to a large extent.

*2.5.3.3 Limitations Of The Model*

The experimental context of this work is the most complete of any empirical study to date. Since the simulated discovery task was less rigidly defined in Reimann's (1990) REFRACT environment, Klahr and Dunbar found that different discovery paths may be followed depending on the state of the investigation and the preferences of the subject.

However, like Reimann's study and model, there has been no investigation of high-level theoretical knowledge that is applicable across several different experimental situations (of the sort called hypotheses in the present framework). Such universal laws are an important part of science. To rectify this, Klahr and Dunbar's robot experiments would need to investigate several different mystery functions. Descriptions for a partial set of the functions would be found (models) and then a general account (an hypothesis) inferred. This account would then be tested by making predictions with the remaining functions.

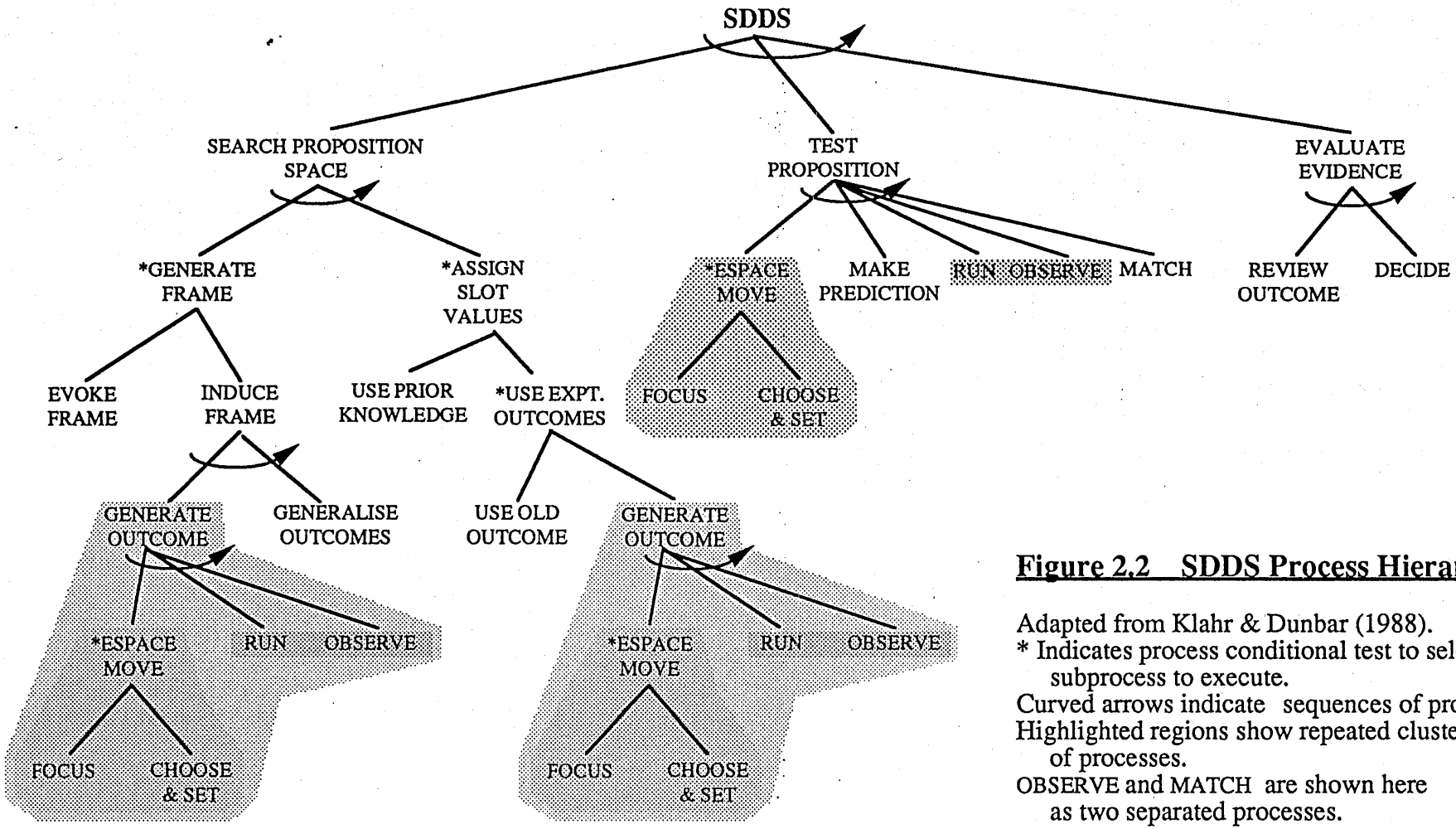



**Figure 2.2 SDDS Process Hierarchy**

Adapted from Klahr & Dunbar (1988).
* Indicates process conditional test to select subprocess to execute.
Curved arrows indicate sequences of processes.
Highlighted regions show repeated clusters of processes.
OBSERVE and MATCH are shown here as two separated processes.

The model is also limited in the way its assesses theoretical acceptability and the reliability of the experiments. SDDS has a location for the evaluation of evidence in its process hierarchy but does not specify what form the assessment of acceptability might take, other than that previous outcomes are reviewed. The robot experimental environment was perfect in the sense that it did not suffer from noise or erroneous effects. Gerwin (1974) included noise in his data which meant human subjects and his computer program often failed to find the exact equation. It would be interesting to see the effect of noise and other adverse influences in future work on SDDS.

### 2.5.4 KEKEDA

Kulkarni & Simon (1988) have simulated Hans Krebs' discovery of the Urea cycle in biochemistry, using their KEKEDA system. This is, perhaps, the most detailed and best model of an episode of scientific discovery to date. The strength of the model comes from the historical account of the discovery at their disposal. This included a detailed examination of laboratory notebooks and retrospective interviews with participants in the discovery. (Again, the use of 'hypothesis' by Kulkarni & Simon is replaced by 'proposition' in this subsection).

#### *2.5.4.1 Representations*

KEKEDA not only makes theoretical inferences, but also assesses the acceptability of the theoretical knowledge and models experiments to a degree. Thus it has representations for theoretical and experimental knowledge and employs measures of acceptability.

KEKEDA works in the domain of biochemistry, and has representations for processes, substances, propositions (ie. "hypotheses"), experiments, and supplementary facts. These representations are classes of attribute-value pairs and contain a rich variety of information. For example, processes are chemical reactions represented by an input, an output, a likely locus of the reaction, and the class to which it belongs. Experiments are also represented in the same manner. An experiment is defined by attributes for: the input; the input's initial values; the

condition and location of the experiment; and indicators of what is to be measured.

Propositions have associated measures of confidence along five dimensions that include: number of successful experiments verifying the hypothesis; number of experiments failing to do so; amount of failed effort in attempts to find positive instances; implied but inconclusive success; and implied but not certain failure. They not only state how acceptable the propositions are, but help to guide KEKEDA in its discoveries. For example, the current proposition is abandoned when the measure of failed effort, of attempts to find positive instances of the proposition, passes a certain threshold. This happens even though there is insufficient evidence to show that the proposition is unacceptable.

*2.5.4.2 Heuristics And Simulation*

KEKEDA is a production system. Sixty four heuristics, productions, are employed and grouped into nine classes according to the type of task performed. The classes are: problem choosers, problem generators, decision makers, experiment proposers, expectation setters, proposition generators, proposition modifiers, confidence modifiers, and proposition-strategy choosers. There are roughly equal numbers of domain specific and domain independent heuristics.

KEKEDA simulates discovery of the urea cycle in some detail, starting with the problem of urea synthesis and working through to the full specification of the cycle, including the pursuit of unproductive paths along the way. During the discovery particular patterns of heuristics repeatedly fire in sequence. For example, when testing the alternative combinations of substances in a particular class of reaction. The program designs experiments and makes predictions, but the user supplies KEKEDA with the results of the appropriate experimental tests when requested.

*2.5.4.3 Criticisms of KEKEDA*

The obvious limitations of KEKEDA are that it does not model the most general levels of theory and experiment - hypotheses and experimental paradigms in the framework. However, we should remember that there was no intention on the part

of Krebs in the real discovery to find a universal law of chemistry, but only to uncover the secrets of a particular biochemical processes.

KEKEDA does, however, possess the most sophisticated means, of any of the systems, to assess the acceptability of propositions (excepting ECHO). The measures that directly indicate the acceptability of a proposition do so by noting the extent to which the proposition successfully predicts the behaviour of the phenomena. This is a further example of the explanatory breadth criterion. Note also that KEKEDA needs to record the absolute number of both successes and failures, unlike Reimann's (1990) HDD program where a single strength value is used. This extra information is important for KEKEDA as it helps to guide the choice of strategies.

The detail and thus completeness of the simulation of the discovery of the Urea cycle is much greater than the level attained in any of the previous models of discovery. This is not surprising as there is an order of magnitude difference in numbers of heuristics employed. There are two implications to be drawn from this. First, it seems that realistic models of episodes of scientific discovery are more likely to be achieved by sets of domain-specific heuristics rather than a single, all encompassing, technique. Second, the modelling of scientific discovery in computer programs will require the investigation of several different domains in detail followed by the generalization of patterns common to each in order to understand the underlying character. A step towards this will be to develop computer models that possess a richness of heuristics and representations comparable to KEKEDA, but implemented in a system with an explicit organization of tasks and processes like SDDS. As we will see the STERN model of scientific discovery attempts such an integration.

### 2.5.5 Summary Of Multiple-Process Models

In this section models that incorporate both theory and experiment led strategies have been examined. They represent and model experiments to varying degrees and employ a range of measures and procedures to assess the acceptability of theoretical knowledge. In fact, its does not seem just to be coincidental that the more experimental knowledge is acknowledged the more realistic the methods for the assessment of the acceptability of theories become. In terms of the scientific discovery framework this is explained by the proposal that the acceptability of models is assessed as a function of the success of experimental tests matching with (predictive) instances. The problem of the acceptability of higher levels of theoretical knowledge, framework hypotheses, has not been addressed in previous work as they simply have not modelled that type of theory.

## 2.6 IMMEDIATE RESEARCH OBJECTIVES

The review carried out in this chapter reveals a number of directions in which the development of computational models may profitably progress. These include:

- The modelling of the highest level of theoretical knowledge to account for phenomena across several different experimental situations (including the processes that generate them and use them to make inferences).
- The realistic modelling of all the types of experimental knowledge, and the processes that manipulate the knowledge, in as much detail as that used to model theories.
- The assessment of the acceptability theoretical knowledge, particularly in terms of the breadth or scope of experimental evidence for which it can account.
- The investigation of the different types of communications that occur between theoretical and experimental components; in particular, modelling the correspondences between theoretical terms and experimental parameters.
- The development of systems with a range and richness of representations and heuristics equivalent to KEKEDA's (Kulkarni & Simon, 1988), but organized by an over-arching

scheme like the process hierarchy of SDDS (Klahr & Dunbar, 1988). This will require the explicit separation of general and domain specific knowledge and rules.

- The initial examination of processes to incorporate experiment reliability measures into the assessment of theory.

The development of future models must also pay attention to the lessons learnt in previous work. This in effect places a number of constraints, or guidelines, on the development of new discovery systems, these include:

- Representing domain specific knowledge using formalisms that are natural and realistic, to avoid problems of vagueness and ambiguous interpretation.
- Acknowledging the importance of both qualitative and quantitative knowledge representations and inferences in discovery.
- The use of BACON-like regularity spotters as an effective way to find relations between terms and quantitative descriptions of data.
- The formation of a close integration between the processes that generate new knowledge and those that assess the acceptability of that knowledge.

The STERN discovery system will take up the suggestions for future development, whilst trying to satisfy the set of constraints. The description of the system begins in Chapter 4, while the intervening chapter considers the episode of discovery modelled by STERN.

# Chapter 3

# Galileo And Natural Accelerated Motion

## 3.1 WHY GALILEO'S DISCOVERIES?

In this Chapter an account of Galileo's investigation of the motion of terrestrial bodies moving under the effect of gravity will be related in some detail. The STERN discovery system models this episode of scientific discovery.

Several characteristics of Galileo's discoveries make the episode a good candidate for modelling. First, experimentation plays an important role in the episode; several different experimental paradigms are employed and even new experimental paradigms are invented. Second, it is a domain that requires a rich interplay of qualitative and quantitative formalisms. Third, since the discovery of the law of free fall has been superficially modelled by BACON the treatment of the episode within the present framework emphasizes the advances that can be made. Fourth, dynamics is a well-established area with adequate laws that account for phenomena with known levels of experimental noise and accuracy. Thus the user will not need to hand calculate experimental outcomes.

The experiments that Galileo used will be described first, followed by a consideration of the way theoretical knowledge and inferences were expressed at the time. Finally, the chronology of discovery events is outlined.

## 3.2 EXPERIMENTS

The central role of experiments in the discoveries of Galileo should not be underestimated. Whereas, previous thinkers, like Aristotle, simply relied on mere observation in their qualitative attempts to characterize motion, Galileo performed experiments on the phenomenon. He manufactured experimental apparatus in which

the motions of bodies could be carefully manipulated and their consequent behaviour accurately measured. This allowed him to form quantitative laws and to test them rigorously. For this reason, Galileo is often be considered to be the first *scientist* in the modern sense of the term.

The most important classes of motion experiments used by Galileo include (Galileo, 1838):

(i) swinging pendulums consisting of small weights attached to the end of long suspended chords set into oscillatory motion (MacLachlan, 1976); and

(ii) inclined planes, or ramps, made from long straight wooden batons down which spherical metallic balls are rolled (Settle, 1961).

Figures 3.1a and b show these two experiments schematically. Although the possibility of performing accurate experiments using such equipment has been doubted (Koyre, 1968), actual reconstructions have shown such claims to be ungrounded (e.g., Settle, 1961).

The empirical side of Galileo's work can be characterized easily by the experimental component of the framework. The simple pendulums and inclined planes are different classes of experiment or *experimental paradigms*. The manufacture of an inclined plane gives an *experimental setup* that has many kinds of parameters (e.g. including: the distance down the plane, its height, the size, weight and volume of the ball). Some of the parameters for each experimental paradigm are shown in Figures 3.1a and b. In the inclined plane paradigm distances can be determined from markings made on the side of the plane. However, the measurement of time was more difficult and required Galileo to use a water clock (and sometimes his own pulse!). Thus certain parameters are easier to control or measure than others, which to a large extent determines the selection of parameters that occupy the particular roles in an experimental test. For example, when using the inclined plane to investigate the relationship between distance and time in an

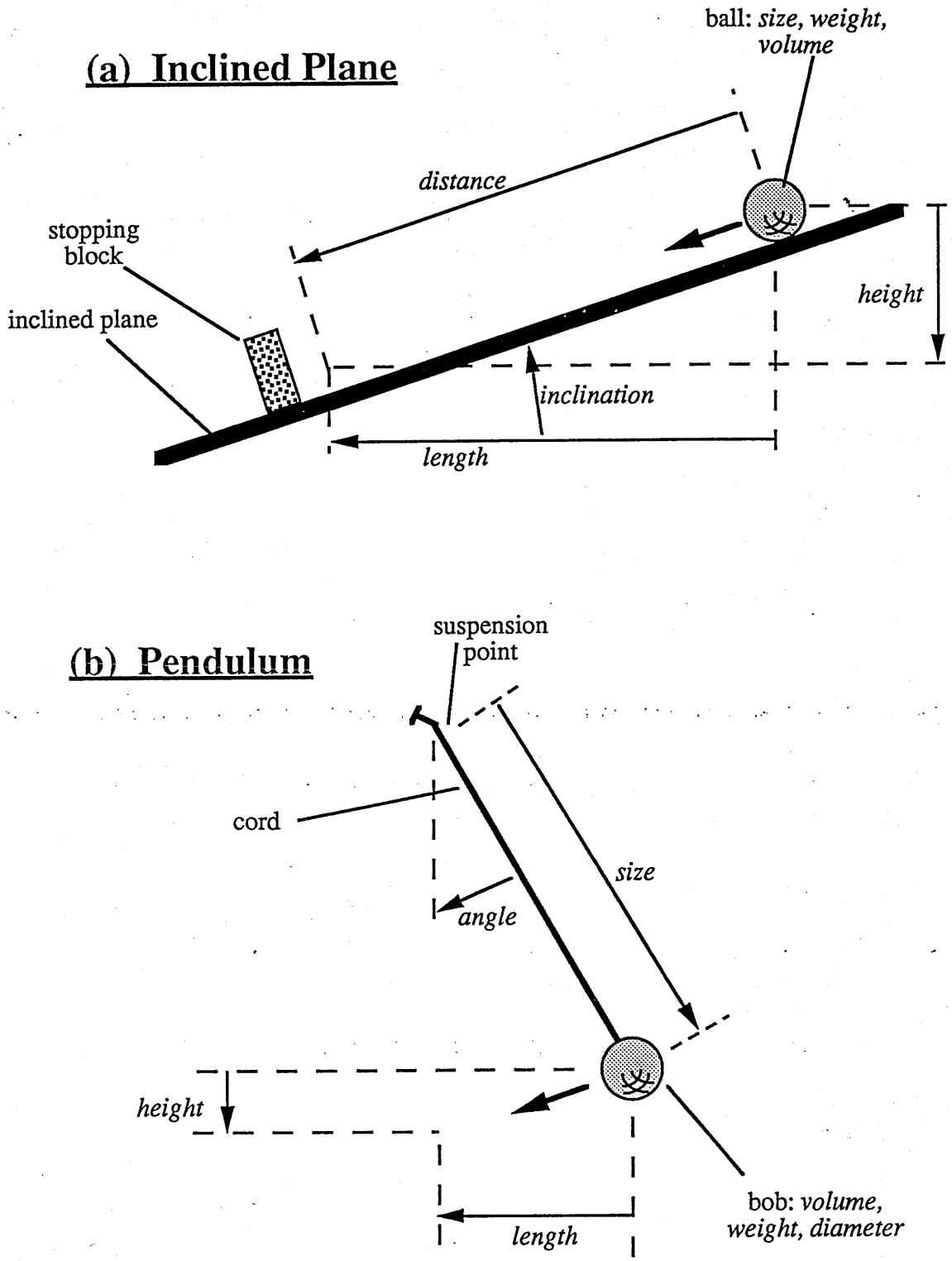



**Figure 3.1 The Inclined Plane And Pendulum Experiments**

*Italics indicates parameters.* Direction of arrows indicates sense of parameters.

*experimental test*, the distance is simpler to manipulate as the *input-m* parameter because a stopping block can be placed on the plane next to specific marks, so forcing time to be the *output* parameter.

Galileo's skill as an experimental scientist is shown by his invention of new experimental paradigms. The basic technique he employed was to combine known experiments using the output of one to feed into another. For example, Figure 3.2 shows the combined projectile and inclined plane experiment (Drake & MacLachlan, 1975; Drake, 1975). In this experiment a ball descends an inclined plane, PQ, and is launched into the air with an imposed initial horizontal motion by the lip at Q, and freely describes a path as a projectile until it lands at R. The first half (inclined plane) of such combined experiments will be called the *initial* part of the experiment, and the second half (projectile) called the *terminal* part. There are two ways (which I call *modes*) in which combined experimental setups can be used in tests, see Figure 3.3 which shows the two parts of combined experiments as black boxes. The *initial* mode (a) employs a parameter from the initial part to be the overall input-m and measures a terminal parameter as the output; for example the height of the inclined plane as the input-m and horizontal projectile length as the output. The *terminal* mode (b) focuses just on the terminal part of the combination, both overall input-m and output being parameters from that part, with the initial part output acting as a fixed terminal input parameter; for example projectile height and length, with fixed inclined plane height. As we will see Galileo carried out investigations on combined experiments in both the initial and terminal modes.

We have already seen how the relative ease with which parameters can be manipulated or observed influences the selection of input-m and output parameters during the design of an experimental test. However, this is not the only form of domain specific knowledge that is associated with the Galilean experiments. The relative ease of manufacture of experimental setups from particular paradigms plays a role in the selection of the setups. A pendulum paradigm is very much simpler to

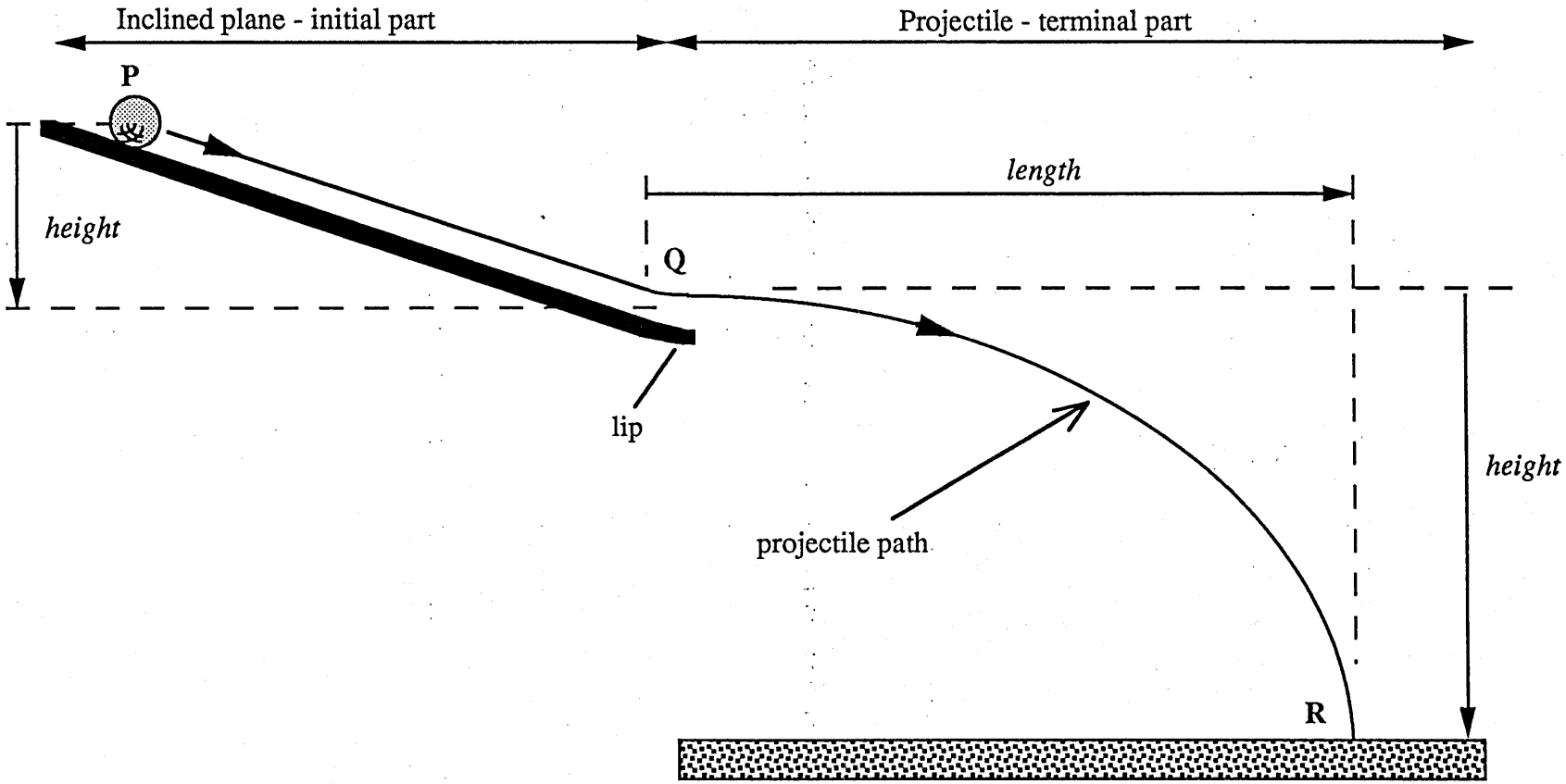
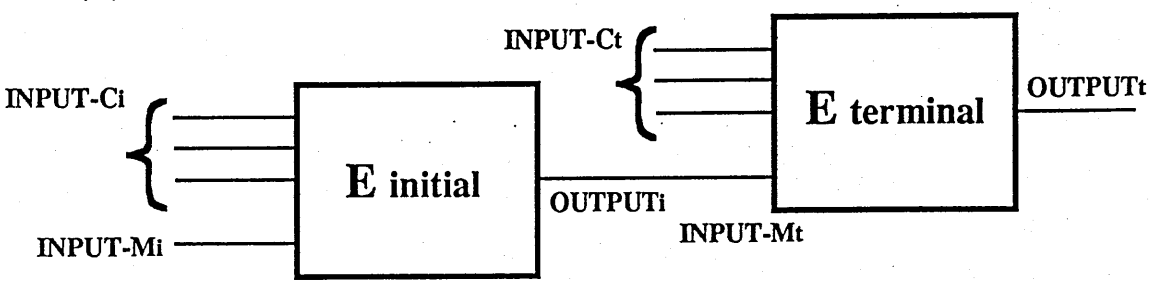


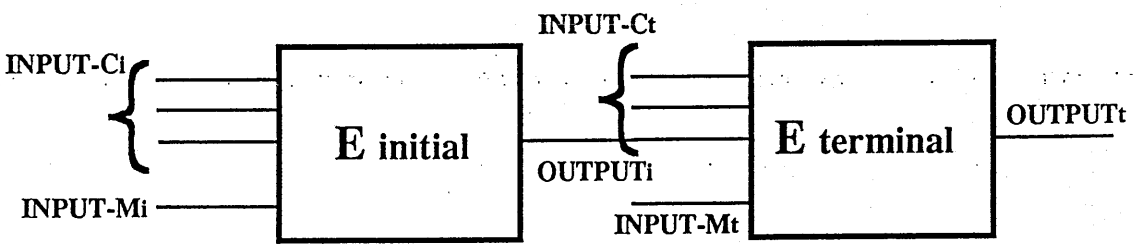

**Figure 3.2 The Combined Inclined Plane And Projectile Experiment**

Key: *Italics indicates parameters.*. Direction of arrows indicates sense of parameters. P - release, Q - *initial* to *terminal* transition, R - landing.

(a) initial model




(b) terminal model




(c) overview

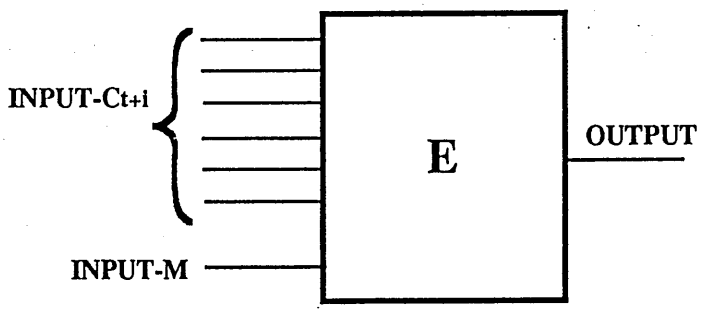



Figure 3.3 Combined Black Box Experiments

construct than an inclined plane.

The design of experiments does not just rely on knowledge that is specific to experiments. Background knowledge also has an important role to play. When choosing the parameters in a given experimental setup to be the input-m and output it is essential to ensure that the parameters are not trivially related. For example, when the inclination of an inclined plane is fixed, the distance, height and length will vary in proportion to each other just because of the geometry of the setup. In a pendulum with a fixed angle the size, height and length are also related together in a similar way. Experiments with such parameters would produce irrelevant or even misleading results. However, simple geometrical knowledge about triangles and circular arcs will tell the experimenter that these parameters are related together in a manner that is completely independent of the phenomenon in the black box. Galileo possessed and used his knowledge of geometry and also knew of relationships between the diameter, volume, and weight of spherical bodies.

The reader may be wondering why there has been no mention Galileo's most famous experiment that involved dropping two balls of unequal weight from the Leaning Tower of Pisa. The case has been omitted because there is no historical evidence to show that any such experiment was ever performed. The origins of this myth seem to be in a *thought* experiment that Galileo conceived as an illustration of his actual empirical findings.

## 3.2 THEORIES AND INFERENCES

On the theoretical side of the investigation, Galileo initially believed that the natural motion of bodies was adequately described by an existing theory (set of so-called laws) that originated from Aristotle. Two important laws were the *instantaneous acceleration law* and the *effective weight law.* The instantaneous acceleration law states that acceleration lasts only for a very brief period at the start of the motion, followed by more or less constant velocity motion. The velocity

attained by a body is in proportion to its *effective weight* according to the second law. Effective weight meaning either weight or density, in modern terms. It is interesting to note that the first is an example of a qualitative law typically expressed propositionally, while the second is a quantitative law that would have been expressed as an equation between ratios of terms, e.g.:

$$w^*_1/w^*_2 = V_1/V_2, \qquad \ldots (3.1)$$

where w* is effective weight, V is speed and the subscripts refer to different bodies or situations. Compared with modern equations in physics, the ratio expressions make manipulations more complicated and limits the form of the equations that can be stated, but avoids the need to consider constants of proportionality. Equation (3.1) is easily categorized as a state transformation function in the present framework.

Galileo carried out several types of theoretical inference. This included the simple manipulation and substitution of terms in equations like (3.1). However, he also employed a geometrical-pictorial method to generate models that is unusual compared to modern conventions (e.g. Drake, 1973a, 1973b; Humphreys, 1967). For example, consider how Galileo inferred the speed of bodies descending an inclined plane from the effective weight law (see, Humphreys, 1967). From the picture in Figure 3.4 he represented the inclined plane by the line *gh*, which is a tangent to a circle with centre *a*. Line *ef* indicates the path of a body falling freely due to its unmodified effective weight. Now, as the effective weight along *gh* is in proportion as *ap* is to *ad*, then the ratio of the speeds of *gh* and *ef* is *ap*:*ad*. That is, for the same height, the greater the angle of the inclined plane, the greater the speed. This geometric-pictorial method is not particularly rigorous, nor easy to use, and only has sufficient expressive power to cope with the simple laws that Galileo considered. It is not surprising that Newton needed to invent infinitesimal calculus before being able to fully develop, state and apply his own theories of

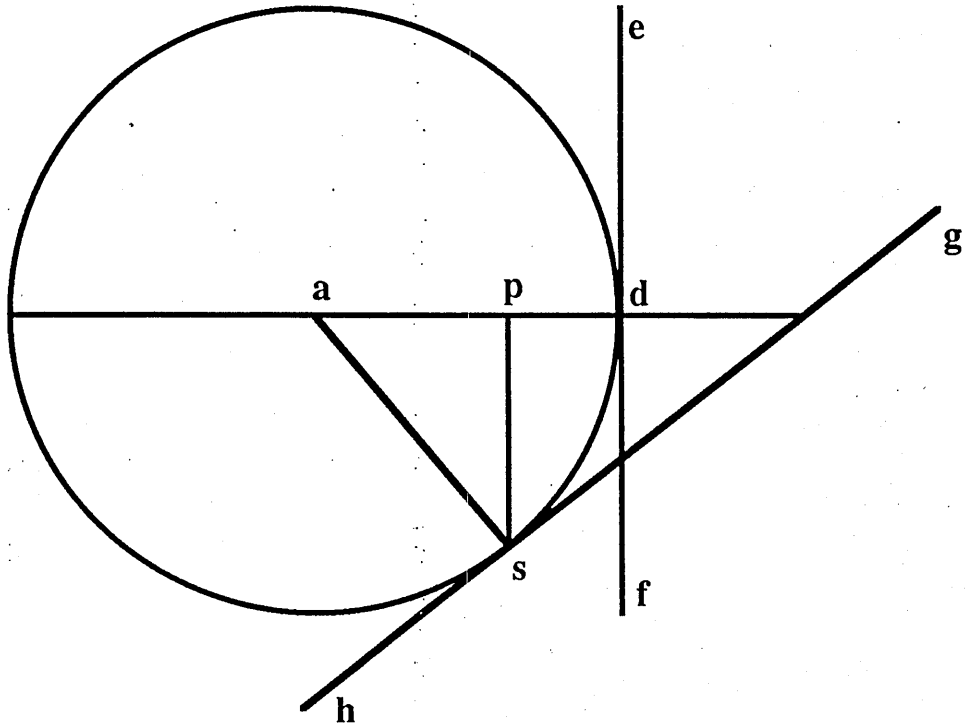



Figure 3.4 An Example Of The Galilean Geometric-Pictorial Method

motion.

## 3.3 CHRONOLOGY OF DISCOVERY

Galileo's discoveries took place between 1590 and the early sixteen-hundreds (see Drake, 1975). From the outset he was interested in all types of bodily motions, but focussed his effort on the smaller and better delimited problem of naturally accelerated terrestrial motion. In modern terms this is motion influenced by the earth's gravitational field. All together his use of both theory and experiments make up the research programme component of the framework.

### 3.3.1 Aristotelian Laws Disconfirmed

Galileo started his investigations by adopting the Aristotelian laws and attempted to test their validity by performing experiments. By careful observation in experiments using long slow swinging pendulums, Galileo saw that the speed of the pendulum bob increased throughout the swing, from its release to the lowest point of the arc (Drake, 1975). This was a direct disconfirmation of the instantaneous acceleration law. The test of the effective weight law was more involved (Humphreys, 1967). Using his characteristic style of geometric inference, Galileo made the prediction from the law that the speed of the balls rolling down inclined planes would be inversely proportional to the length of the plane. However, actual inclined plane experiments showed that the speeds were independent of the length of the plane. By checking these findings across the available experimental paradigms, Galileo obtained sufficient confidence to abandon the Aristotelian views of motion.

### 3.3.2 Finding Laws

During the disconfirmation of the Aristotelian laws, Galileo would have gathered many experimental results that were not specifically used in the disconfirmation process. Further, once the laws had been abandoned Galileo continued to explore the phenomenon using the various experimental setups that had

been constructed. Thus qualitative and quantitative characterizations of the nature of the phenomenon were formed by generalizing the gathered experimental results that applied to specific experimental situations. Individually or in groups, some of these models were further generalized to form hypotheses.

Naturally, these hypotheses were then tested by comparison with experimental results in the manner that the effective weight and instantaneous acceleration laws had been considered.

### 3.3.3 Proposing New Hypotheses

Galileo had thus built up a wealth of information about the phenomenon and was in a position to generate quantitative hypotheses, including the law of free fall. In the inference to the law of free fall Galileo first made the assumption, based on existing knowledge, that distance increases with the natural numbers (1,2,3,...) as the speed increases with the odd natural, numbers (1,3,5,...). Then by a mixture of qualitative and quantitative manipulations, using the geometric-pictorial form of reasoning, he found (or rather stumbled upon) a simple relation between distance and speed (Drake, 1973a, 1973b). The relationship stating that the velocity squared is proportional to the distance travelled (or distance travelled is proportional to the square of the time), ie:

$$V_1^2/V_2^2 = d_1/d_2, \qquad \ldots (3.2)$$

where $V$ and $d$ are speed and distance, respectively, and the subscripts relate to different distances on the inclined plane. Other hypotheses were also considered, such as, that velocity was linearly related to the distance travelled. To find which new hypotheses were correct further experiments needed to be performed.

### 3.3.4 Inventing Experiments To Test New Hypotheses

To experimentally test the law of free fall Galileo had to invent a new experiment, because of a problem concerning the speed term. Galileo wished to eliminate the term from the free fall equation and replace it with other terms that could be directly measured in experiments. He knew that speed was defined by the

ratio of distance over time but only when the speed was constant, i.e. no acceleration. So distance and time terms could replace the speed term if the actual speed in the experiment was constant. Previously, when attempting to confirm the Aristotelian laws, Galileo thought the instant acceleration law, which stated that velocity was constant for most of the time, was true and was thus happy to make the substitution. However, by now it had been found that gradual acceleration occurs in both the inclined plane and pendulum motions. Galileo's solution to this problem was ingenious. He developed a new combined experimental paradigm by using the inclined plane as a launcher for projectiles (Figure 3.2). Galileo knew that the horizontal velocity of a projectile was constant and he had the law of free fall describing the speed of the ball at the end of the ramp. Thus considering the combined experiment in the initial mode, he was able to substitute out the speed from both equations and obtain a relation between the height of the inclined plane and the horizontal distance travelled as a projectile. As both parameters were measurable quantities in the experiment Galileo was able to perform experiments that confirmed the law of free fall (Naylor, 1974).

Whilst performing those experiments Galileo was able to closely observe the flight of a projectile for the first time, and thus became interested in the shape of the path described in this motion. By applying the newly-confirmed law of free fall to projectile motion he was able to make predictions about its exact trajectory. These predictions were confirmed by further experiments using the original setup in the terminal mode. This not only explained the shape of the path but also increased the acceptability of the law of free fall (Naylor, 1975; Drake & MacLachlan, 1975; Naylor, 1976; Hill, 1988). The shape described by a projectile is parabolic, the horizontal length, L, increases with the square of the vertical height, H; ie

$$H_1^2/H_2^2 = L_1/L_2, \qquad \ldots (3.2)$$

where the subscripts refer to different points on the path.

### 3.3.5 Switching Research Programs

Galileo's work on dynamics stopped around this time his attention being diverted when he learnt of the invention of the telescope. Galileo used the new instrument to make other discoveries for which he is also justly famous. In the *Dialogues Concerning Two New Sciences* Galileo (1838, 1954) gives a full account of his dynamical theories, although little is said about the way in which the law of free fall was really discovered. It is well known that Newton took up, more or less, where Galileo left off and developed laws of motion that superseded the law of free fall. However, it is not so widely acknowledged that Newton also inherited Galileo's experimental legacy, and performed experiments that were adaptations of Galilean paradigms.

## 3.4 CONCLUSIONS

The Galilean episode of discovery will be modelled by the STERN discovery system, so it seems appropriate to make a few relevant observations.

- Qualitative and quantitative theoretical knowledge had complementary roles in the episode.
- Galileo did not consider theories that had been generalized from experimental results to be inherently true, but tested them against other experimental paradigms before accepting them.
- In the generation of predictive models, attention was paid to the particular experimental paradigm that was being modelled and the terms specified in models corresponded to measurable and observable parameters of an experimental setup.
- Several different experimental paradigms were used by Galileo and he had pragmatic knowledge concerning each one: for example the relative ease with which experimental setups could be manufactured, and the ease with which particular parameters could be observed.
- Four general discovery processes can be identified: (i) the confirmation of existing

hypotheses (§3.3.1 & §3.3.2); (ii) the generalization to new hypotheses (§3.3.2); (iii) the formation of new hypotheses from existing ones (§3.3.3); and (iv) the invention of new experiments (§3.3.4) permitting further hypotheses to be tested.

- Background knowledge played a significant part in the inferences, in particular it helped to identify parameters that were trivially related through the geometry of specific experiments and it was used to make predictions from hypotheses.

It is interesting to note that many of these points overlap with the requirements and proposals for computational discovery systems outlined in the conclusions to Chapter 2 on previous work.

We have seen how Galileo made important scientific discoveries. In the remaining chapters of this thesis we will also see how the STERN discovery system also successfully models the discovery of the same models and hypotheses using a range of experimental paradigms.

# Chapter 4

## STERN: Scientific Theorist And Experimental Researcher

### 4.1 INTRODUCTION

In the remainder of this thesis we apply the framework (described in Chapter 1) to Galileo's discoveries on the motion of naturally accelerated bodies (see Chapter 3). The result of this work is the STERN (N is the version number, currently 0) computer program which is described in the remaining chapters. STERN attempts to overcome many of the limitations of previous computational models of scientific discovery (reviewed in Chapter 2).

STERN has several notable abilities, which can be summarized as follows:

- it considers all the types of theoretical and experimental knowledge posited by the framework, including hypotheses and experimental paradigms, using frame like representations in a well-ordered hierarchical structure.
- it possesses a wide variety of processes and heuristics that can apply existing knowledge, or infer new knowledge.
- it distinguishes explicitly between domain-specific and domain-independent classes of processes, which are further organized in groups in a task hierarchy.
- it implements processes as condition-action rules in a production system architecture.
- it instantiates of multiple types of information transfer between theory and experiment, including the correspondence between theoretical terms and experimental parameters.
- it employs quantitative and qualitative representations and has the ability to make

quantitative and qualitative inferences alone or in combination.

- it assesses the acceptability of theoretical knowledge by an adequacy function based on the relative success of hypotheses, models and instances.
- it considers methods to deal with noise in experimental results.

Although STERN is based on the framework, the range of different scientific domains it can model is smaller than that covered by the framework itself. In its present form, STERN can only cope with episodes of discovery that have similar types of hypotheses and models to those found in the Galilean case (i.e. mainly quantitative). Even so, this means that STERN is able to encompass a significant number of important scientific disciplines. The application of the framework to other scientific domains will provide interesting future research.

This chapter is the first of five describing STERN and the way it models the Galilean episode. The following chapters, 5 to 8, consider the main types of processes that STERN uses at different times when making its discoveries. The present chapter focuses upon basic features of the program and gives an overview of its performance. The following sections consider: (§4.2) an overview of the discoveries made and the path followed whilst making them; (§4.3) the general instantiation and representation of the components of the framework for scientific discovery; (§4.4) the representations needed to model the knowledge specific to the Galilean domain, including pertinent background knowledge; (§4.5) the nature of STERN's heuristics, their organization as a hierarchy, and their implementation in a production system; (§4.6) the subprogram that simulates real experiments; and (§4.7) STERN's top level of control, that chooses different discovery strategies.

The program is implemented in Common LISP (CL) and runs on an Apple Macintosh II.

## 4.2 AN OVERVIEW OF STERN'S DISCOVERIES

STERN models Galileo's discoveries in the domain of naturally accelerated motion. This involves the interaction of many types of theoretical and experimental knowledge over a great many cycles of the program. This section presents an overview of STERN's performance that will help to set into context the much more detailed considerations to follow in this and later chapters. We will start with inputs and outputs of the program.

### 4.2.1 Inputs & Outputs

STERN is given the three hypotheses and six experimental paradigms. The laws are the instantaneous acceleration law and two interpretations of the effective weight law (section I, Table 4.1[1]). T_V, T_D, T_DEN and T_W* are theoretical terms standing for speed, distance, density, and effective weight. As we will see below, STERN represents qualitative relations using *qualforms*; the instantaneous acceleration law is a typical example. The acceptability of this hypothesis is set to a moderate value to reflect the fact that Galileo initially thought it to be true. STERN is told that the two versions of the effective weight law have not been examined before to acknowledge that Galileo was the first researcher to investigate them quantitatively. Galileo's most important experimental paradigms were the pendulum and the inclined plane, so STERN is given representations of them both and told they have been manufactured. STERN is also given knowledge about other experimental paradigms; for instance, projectile experiments. These paradigms are not yet available for use but just known about conceptually by STERN.

Two sets of background knowledge are provided to instantiate a knowledge of geometry and relations for spherical bodies.

What does STERN discover? (i) It finds that the three Aristotelian laws are unacceptable. (ii) STERN obtains a thorough qualitative understanding of the

[1]The meaning of all the different entries of this table will become clearer as we progress through this chapter.

Table 4.1 State of STERN's Hypotheses After 1600 Cycles

| Section | Equation or Qualform | Adequacy† | Tractability† | Grouping |
|---|---|---|---|---|
| I | (= T_V T_DEN) | 2 0.322 | 2 2 | |
| | (= T_V T_W*) | 2 0.000 | 2 2 | |
| | (INSTANTANEOUS T_V T_D) | 3 1 | 3 3 | |
| II | (INCREASE T_TIME T_H) | 2 2 | 2 2 | |
| | (FROM_ZERO T_TIME T_H) | 2 2 | 2 2 | |
| | (INCREASE T_TIME T_L) | 2 2 | 2 2 | |
| | (FROM_ZERO T_TIME T_L) | 2 2 | 2 2 | |
| | (STEADY T_TIME T_W) | 2 2 | 2 2 | |
| | (STEADY T_TIME T_VOL) | 2 2 | 2 2 | |
| | (INCREASE T_V T_H) | 2 2 | 2 2 | |
| | (FROM_ZERO T_V T_H) | 2 2 | 2 2 | |
| | (INCREASE T_V T_L) | 2 2 | 2 2 | |
| | (FROM_ZERO T_V T_L) | 2 2 | 2 2 | |
| | (STEADY T_V T_W) | 2 2 | 2 2 | |
| | (STEADY T_V T_VOL) | 2 2 | 2 2 | |
| III | (= T_S (* T_TIME T_TIME)) | 1 1 | 3 1 | |
| | (= T_D (* T_TIME T_TIME)) | 2 1 | 2 2 | |
| | (= T_H (* T_TIME T_TIME)) | 2 1 | 2 2 | |
| | (= T_L (* T_TIME T_TIME)) | 2 1 | 2 2 | |
| IV | (= T_V (EXPT T_H 1/2)) | 2 1.862 | 4 2 | GROUP3541 |
| | (= T_V (EXPT T_H 1/3)) | 1 0 | 2 0 | GROUP3541 |
| | (= T_V (EXPT T_H 2)) | 1 0 | 2 0 | GROUP3541 |
| | (= T_V (EXPT T_H 2/3)) | 1 0 | 2 0 | GROUP3541 |
| | (= T_V (EXPT T_H 3)) | 1 0 | 2 0 | GROUP3541 |
| | (= T_V (EXPT T_H 3/2)) | 1 0 | 2 0 | GROUP3541 |
| | (= T_V T_H) | 1 0 | 2 0 | GROUP3541 |
| V | (= T_V (EXPT T_L 1/2)) | 2 0.000 | 4 2 | GROUP3542 |
| | (= T_V (EXPT T_L 1/3)) | 2 0.000 | 4 2 | GROUP3542 |
| | (= T_V (EXPT T_L 2)) | 2 0.000 | 4 2 | GROUP3542 |
| | (= T_V (EXPT T_L 2/3)) | 2 0.000 | 4 2 | GROUP3542 |
| | (= T_V (EXPT T_L 3)) | 2 0.000 | 4 2 | GROUP3542 |
| | (= T_V (EXPT T_L 3/2)) | 2 0.000 | 4 2 | GROUP3542 |
| | (= T_V T_L) | 2 0.000 | 4 2 | GROUP3542 |

† - 1st and 2nd number are the fillers of the number and degree slots of the measure frame respectively.

domain; finding which terms are relevant to the characterization of the phenomena (e.g., the weight of bodies is irrelevant but height is significant). (iii) Some quantitative models (not hypotheses) are also discovered; for example the law governing the period of swing of pendulums. (iv) STERN discovers that the law of free fall is the only acceptable quantitative hypothesis from amongst many proposed. (v) New experimental paradigms are constructed; for example the combined inclined plane and projectile experiment.

Let us consider in a little more detail how STERN made all these discoveries.

### 4.2.2 Discovery Path

Figure 4.1 shows, at a very coarse grained level of detail, the various stages that STERN progresses through when modelling the Galilean episode of discovery. STERN chooses to perform confirmation of known hypotheses four times during the whole run of the program. Confirmation first occurs when the three Aristotelian laws are given as input and then later after STERN has (i) generalized experimental results into hypotheses, (ii) generated new hypotheses from old, and (iii) considered new experiments.

STERN attempts to confirm the Aristotelian laws by considering each with respect to the pendulum and inclined plane experimental paradigms in turn. For each hypothesis and paradigm pair, STERN attempts to account for the phenomenon in the paradigm by generating models from the hypothesis. STERN disconfirms the instantaneous acceleration law by discovering that the predictions made from the law, when applied to the two paradigms, simply do not hold (i.e. the motion is gradual acceleration). During the disconfirmation of the effective weight laws STERN generates models with equations from the equations of the two laws. This involves the use of the background knowledge of geometry and spherical body relations to replace the wholly theoretical terms (such as *speed* and *effective weight*) with measurable terms (like *distance*, *time* and *weight*). STERN makes

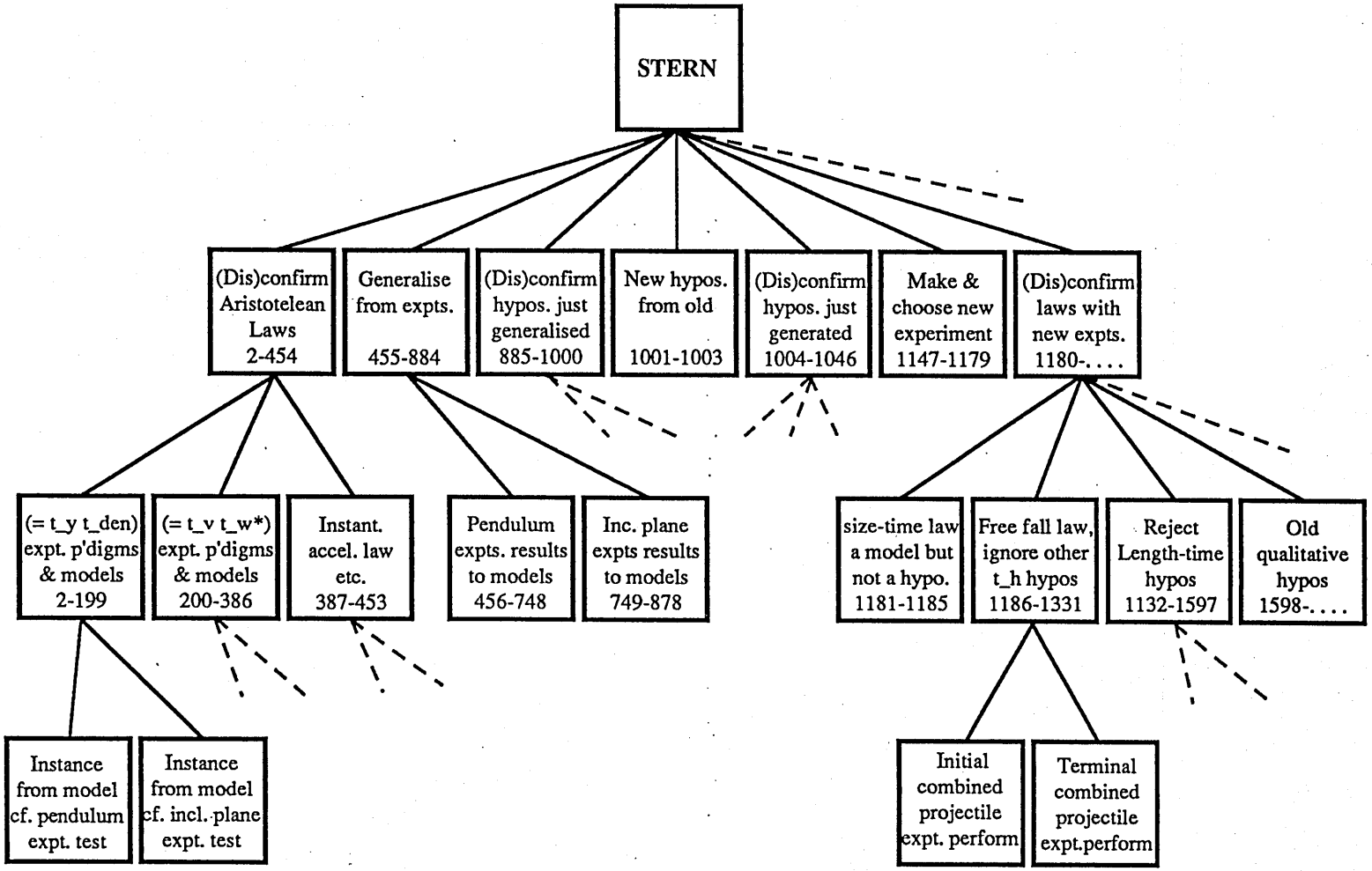



**Figure 4.1 Progress Of STERN's Discovery**

**Notes:**
Numerals within nodes are PS cycle numbers (see Appendix I).
Dashed lines indicate similar sets of processes occur.

predictive instances from the models and compares them with the results of experiments that it designs and performs. To compare the instance and experimental test STERN uses a method that both assesses how closely they match and takes into account any noise that exists in the experimental data. Even so, few of the instances are found to compare well with the experimental results. Thus the models are not acceptable; and in turn, STERN finds the hypotheses are not acceptable either. This is correct because the *effective weight* and *density* terms, referred to by the laws, do not influence the rate of acceleration in naturally accelerated motion.

STERN then decides to obtain further experimental results that can be generalized into hypotheses via instances and models. STERN designs all the possible experiments that can be performed using the pendulum and inclined plane experimental paradigms. Background knowledge is used to eliminate all those experiments that would yield trivial results; such as, increasing the length of a pendulum increases the distance that the bob travels. STERN performs all these sensible experiments and interprets the results. Many quantitative and qualitative instances are found. These instances are then generalized to form models. In particular, the generalization to quantitative models involves finding equations from the lists of independent and dependent values (of the instance), whilst taking into account the presence of noise in the data. It is at this point that STERN discovers the law governing the period of swing of pendulums in terms of cord length (*size*). This is arguably a "genuine" discovery (rather than a modelled rediscovery), because it is a true law found by STERN, but the programmer had not specifically intended that STERN would find this law. It was only with hindsight that the law was seen to be a reasonable possibility given STERN's input. Other quantitative hypotheses were also found (III, Table 4.1). The many qualitative hypotheses found by STERN at this stage provide STERN with a much deeper understanding of the nature of the phenomenon (II, Table 4.1). For example, the qualitative hypotheses indicate that the weight of a body does not affect its acceleration and that

the acceleration is a gradual process in which the height of the body is significant.

Like Galileo, STERN does not simple believe new hypotheses to be true by virtue of being generalized from experimental results. STERN attempts to confirm them to make sure. But, only the quantitative hypotheses are considered as the qualitative ones were generalized from both experimental paradigms. However, none of the quantitative hypotheses are found to be any more acceptable. Furthermore, the law of the period of swing of pendulums is found to be just a model (i.e. only account for one experimental paradigm), because the inclined plane does not have a parameter that is equivalent to the size term of the pendulum paradigm

The range of hypotheses that STERN now possesses is extensive and covers the whole spectrum of acceptability (I, II & III, Table 4.1). STERN considers the generation of new hypotheses from its established stock. The qualitative hypotheses indicate which terms are significant and the general form that equations might take. The unacceptable quantitative hypotheses rule out certain terms and forms of equations. Thus, STERN is able to infer new hypotheses and successfully generates two groups of new hypotheses (IV & V, Table 4.1). The hypotheses have exponential equations that focus on the relation between speed (T_V) and height (T_H), and speed and length (T_L). The first equation of section IV in Table 4.1 is the correct law of free fall, but STERN does not know this yet.

Once again there are new hypotheses. STERN applies the confirmation strategy again, but none of the new untested hypotheses can be used to account for the inclined plane and pendulum experimental paradigms. The new hypotheses have *speed* terms that cannot be eliminated because the definition of speed can no longer be used to substitute out the *speed* term. The definition can only be applied when the speed is constant, but STERN has already established that in both experimental paradigms it is not constant. Thus the measures of tractability of both groups of hypotheses are simply amended.

Since there are hypotheses to be tested but no suitable experimental paradigms, STERN chooses to consider new experimental paradigms. Like Galileo, the approach adopted is one of combining known paradigms together. Six combined experiments are made including the *inclined plane and projectile* paradigm (see Figure 3.2). To avoid having too many experimental paradigms to consider at any time, STERN limits the paradigms that are considered by calculating a pragmatic limit, in terms of the product of number of experimental setups and ease with which setups can be manufactured. Once all the paradigms above a given level of this pragmatic quantity have been exhausted, the value can be reduced until one (or more) other paradigm comes into play (as if it had actually been constructed). Now, in the present case with the new combined experiments, it is the inclined plane and projectile experiment that is selected for use by STERN.

For the last time STERN calls upon its confirmation strategy. STERN just happens to consider the law of free fall first. STERN applies the hypothesis to the combined experiment and successfully generates models for the experiment in both the initial and the terminal models. Further, these models are found to be acceptable by the comparison of their instances and experimental tests. In turn the hypothesis is considered acceptable; STERN can now be considered to have discovered the law of free fall. However, as the free fall hypothesis is acceptable, the rest in the same group (IV, Table 4.1) are, therefore, unacceptable. STERN sets their measures of acceptability to indicate this fact so that they will be ignored in the future. STERN now turns its attention to all members of the other set of hypotheses (V, Table 4.1). They are all found to be unacceptable.

The modelling stops after 1600 cycles of the program. A great number of hypotheses have been considered (Table 4.1). The Aristotelian laws have been disconfirmed; a range of qualitative hypotheses induced; a "genuine" discovery made by finding the law governing the motion of a pendulum as a model; and the

law of free fall shown to be the only adequate quantitative law amongst the many produced. Throughout the episode experimental paradigms and setups have been selected for use, and experimental tests designed and performed, both to confirm theories and as a means to formulate new theories. (Appendix I is a condensed trace of STERN's output.)

### 4.2.3 Comparison With The Real Episode

As the overview demonstrates, STERN has successfully modelled the Galilean episode of discovery. The order of the stages in STERN's discovery match the phases of Galileo's described in Chapter 3. The disconfirmation of the Aristotelian laws occurs in both, followed by the performance of experiments to obtain generalizable results that are tested against existing experiments. STERN and Galileo both used the knowledge gained from their previous explorations of the domain to infer new hypotheses. This in turn requires the invention of new experiments, including the combined inclined plane and projectile experiment, undertaken by both. Finally, the correct law of free fall is found, by both, by testing the most recently generated hypotheses using the combined experiment. Thus at this level of description there is a good match between the route STERN took and the Galilean path to make the same discoveries.

We have seen the many discoveries that STERN has successfully made. It is now time to consider the details of how STERN did this. We will begin by considering how STERN represents scientific knowledge in general, using frames to instantiate all the levels of knowledge in the present framework (see Chapter 1).

## 4.3 INSTANTIATION OF FRAMEWORK COMPONENTS

The framework specifies the minimum set of components required for modelling scientific discovery. The structure of knowledge in STERN follows the framework closely. In general, all types of experimental and theoretical knowledge in STERN are implemented in frame like representations (*à la* Minsky, 1975).

### 4.3.1 Research Programs

In the framework, the research programme component plays only a very minor part at present, in bringing together theory and experiments to work on a particular domain. In the future investigations of cooperative and competitive programmes I expect the notion to become more significant. A prototype of STERN had a separate frame to represent research programmes with slots for theory and experiment, but as the Galilean episode concerns just a single scientist working more or less in isolation, it merely added an extra layer of representation that had no substantive role. Thus the current version of STERN has theory and experiment as its topmost level of knowledge.

As we saw earlier in Figure 1.3, the different levels of theory and experimentation can be viewed as a hierarchy. In STERN each level is represented by a frame and the hierarchy is reproduced by providing slots in each frame for items on the next lowest level in the hierarchy. For example, the *model* frame has an *instance* slot that is filled by the *instance* frames associated with that model. The various representations will be considered in turn.

### 4.3.2 Theory

There are three types of theoretical knowledge in the framework - hypotheses, models and instances. However, we will first consider the theory frame that instantiates the theoretical side of a research programme.

#### *4.3.2.1 Theory*

In STERN theoretical knowledge comes in two forms, declarative and procedural. In other words, what is known and how (domain-specific) inferences are made. Of the ten slots in the theory frame (see Table 4.2) three slots contain declarative knowledge, and the remaining seven procedural knowledge.

The first three slots contain knowledge covering the whole the domain. The *hypos* slot is filled by a list of hypotheses. The *tvars* slot lists all the symbols standing for theoretical terms in the domain. The terms are themselves *T!* frames

Table 4.2 Theory Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| hypos | list of hypothesis frames | Aristotelian/Galilean hypotheses |
| bgknow_relate | relations for theoretical interpretation of background knowledge | entry frames for geometric and bodily knowledge |
| tvars | list of all theoretical terms | symbols standing for term frames |
| generate_hypos | rules to infer new hypothesis form existing ones | *New Hypotheses* |
| generate_models | model generation rules | *Generate models* |
| generate_instances | instance generation rules | *Generate instances* |
| generalise_models | model generalisation rules | *Generalize models* |
| generalise_instances | instance generalisation rules | *Generalize instances* |
| interpret_expttests | rules to interpret experimental test results | *Interpret* |
| instance_vs_expttest | rules to compare instances and experimental test results | *Compare* |

Table 4.3 Hypothesis Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| equation | quantitative domain specific formalism | an equation |
| qualform | qualitative domain specific formalism | a qualform |
| adequacy | hypothesis acceptability | measure frame |
| tractability | hypothesis tractability | measure frame |
| models | models accounted for by hypo. | list of models frames |
| partial_forms | unsuccessfully generated models | term* frame |
| exptnames | names of experimental paradigms that have been considered | various experimental paradigms |
| group | group membership identifier | a symbol common |

(see §4.3.4.2). The *bgknow_related* slot stores a special list that permits sets of theoretical terms to be mapped onto background knowledge (see §4.4.2 below).

The names of classes of domain-specific rules that can manipulate particular types of theoretical knowledge fill the other seven slots. For example, the *generate_hypos* slot contains the class name of the rules that infer new hypotheses from existing ones (i.e. *new hypotheses* in the current versions of STERN). (In §4.5 below the tasks performed by each of the classes of rules are briefly described and discussed in detail in Chapters 5 to 8). The contents of these slots are accessed by STERN's domain-independent heuristics when domain-specific inferences are required.

*4.3.2.2 Hypotheses*

Hypotheses are the most general or abstract type of knowledge that actually accounts for a set of phenomena. They are represented by frames with eight slots (see Table 4.3). The *equation* and *qualform* slots store quantitative and qualitative formalisms respectively. Models are often partial instantiations of a particular hypothesis so are contained as a list in the *models* slot. The names of the experimental paradigms that the hypothesis has attempted to account for are stored in the *exptnames* slot. The ease with which a hypothesis can be applied (e.g., used to generate models) is contained in the *tractability* slot and the acceptability of the hypothesis contained in the *adequacy* slot. Both tractability and acceptability quantities are values calculated by particular discovery processes and represented as *measure* frames (see §4.3.2.5). When attempts to generate models fail, the results are store in the *partial_forms* slot as a list of *term** frames, and may be recalled for later use. Hypotheses that are related together for some reason (e.g., all generated at the same time in one processes) have a common symbol stored in their *group* slots.

*4.3.2.3 Models*

A model frame is similar to the hypothesis frame in several ways (Table 4.4). It

Table 4.4 Model Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| equation | quantitative domain specific formalism | an equation |
| qualform | quantitative domain specific formalism | a qualform |
| exptsetups | names of experimental setups associated with the model | the list of names |
| adequacy | model acceptability | measure frame |
| tractability | model tractability | measure frame |
| instances | instances derivable from model | instance frames list |
| exptparadigm | name of experimental paradigm modelled | name |
| expttype | indication of special types of experiment | symbol specifying terminal, initial, nil |
| not_tested | experiment setup names not accounted for by model | the names |

has equivalent *equation*, *qualform*, *adequacy* and *tractability* slots, and the *instances* and *exptsetups* slots are like the hypothesis *model* and *exptnames* slots. Models are not as general as hypotheses and account for the manifestation of the phenomenon in one situation, thus each frame has an *exptparadigm* slot for the experimental paradigm (situation) that it covers. As we saw in Chapter 3, combined experiments can be used in one of two different modes which the model notes by an appropriate symbol in the *expttype* slot. When STERN cannot use a model to account for a particular experimental setup, or when no experiments can be performed, the name of the experimental setup is stored in the model's *not_tested* slot for future reference.

*4.3.2.4 Instances*

Instances are typically the predictions made from models or the interpreted results of experiments. The instance frame has seven slots that contain specific theoretical terms, their values and associated information (Table 4.5). The *dep* and *indep* slots contain the dependent ("measured") and independent ("manipulated") terms respectively; and the *depvals* and the *indepvals* slots store list of their values. The *other_terms* slot contains the terms with fixed values as a list. Not all instances are quantitative, qualitative instances use a *qualform* slot to store qualitative formalisms. The *degree* slot contains a number (in the range [0 1]) that indicates the acceptability of the instances.

*4.3.2.5 Acceptability And Tractability*

The framework does not specify what form the assessment of the acceptability should take and previous work has shown explanatory breadth to be a primary criterion. Thus, in STERN a simple adequacy function calculates acceptability in terms the range of experimental results successfully accounted for. Experimental test results are compared with instances and the degree of acceptability of the instance is calculated by particular functions (in the *compare* and *interpret* classes of rules, in Chapters 5 and 6). The acceptability of a model is given by the quotient

Table 4.5 Instance Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| dep | dependent term | a term (T! frame) |
| depvals | dependent term's values | list of values |
| indep | independent term | a term (T! frame) |
| indepvals | independent term's values | list of values |
| qualform | qualitative observation | a qualform |
| degree | acceptability of instance | $0 \leq$ real number $\leq 1$ |
| other_terms | other fixed value terms | list of terms (T!s) |

of (i) the sum of instance acceptability values (i.e., the value in the *degree* slot of the *measure* frame contained in the *adequacy* slot of each instance) and (ii) the total number of instances that are accounted for by the model that have been experimentally tested. Hypothesis acceptability is in turn assessed in a similar manner, but with respect to models; specifically, the acceptability is the quotient of (i) the sum of model acceptability values and (ii) the total number of models that are accounted for by the hypothesis. The adequacy of hypotheses and models can thus be summarized by the equation:

$$\text{Acceptability of K} = \frac{\sum \text{k acceptability}}{\text{Number of ks}}, \qquad \ldots (4.1)$$

where K is a hypothesis (or a model) that accounts for one or more models (or instances), k, respectively. All acceptability values range between 0 and 1.

STERN also calculates the tractability of hypotheses and models by recording the success with which models or instances have been generated. Tractability is defined in terms of the number of models (or instances) successfully accounted for by a hypothesis (model) compared to the number of attempts made to form models (instances). This measure plays no part in the calculation of the acceptability of theoretical knowledge but helps to determine the route taken in the discovery path.

The quantities that represent tractability and acceptability are represented as *measure* frames (Table 4.6). Measure frames have two slots; a *number* slot filled by the number of knowledge items or inference attempts, and a *degree* slot that contains the sum of the particular values calculated by STERN. The assessment of both acceptability and tractability occurs incrementally, with the quantity being amended each time a new model (or instance) is considered by STERN. Thus, the values in the degree and number slots related directly to the numerator and denominator of Equation 4.1, respectively.

The use of tractability and acceptability functions makes STERN similar to, but

Table 4.6 Measure Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| number | number of items or events | non negative integer |
| degree | sum of acceptability or tractability values | 0 ≤ real number |

also different from, ECHO (Thagard, 1989a). STERN is similar to ECHO in its use of explanatory breadth, but differs in its incremental assessment. There is a fairly close similarity between KEKEDA's (Kulkarni & Simon, 1988) five measures of confidence and STERN. Both assess acceptability in terms of the relative support by experimental evidence and use pragmatic measures of the effort expended to find the evidence. Unlike PI (Thagard, 1988a) STERN does not use a measure of simplicity in its acceptability function.

It is our assumption that this treatment of acceptability and tractability is sufficient for successful scientific discovery. This can be viewed as something which will be tested by applying STERN to the Galilean case.

### 4.3.3 Experiments

One of the main differences between STERN and previous discovery systems is the extent to which experiment has been modelled. STERN has representations for each of the framework levels of experimental knowledge.

#### *4.3.3.1 Experiment*

Like the theory frame, the experiment frame also has slots containing declarative and procedural knowledge, but they are fewer in number (see Table 4.7). The *exptparadigms* slot contains the domain's experimental paradigms. The other two slots contain the names of the sets of rules that perform experiments (§4.6) or invent new experiments (see Chapter 8).

#### *4.3.3.2 Experimental Paradigms*

Experimental paradigms have nine slots divided into two groups depending upon whether they are generally applicable to all experiments or specific to combined experiments. Consider the former group first (Table 4.8). The generally applicable slots include: (i) a *name* slot that holds the name of the paradigm (e.g., 'incplane' and 'pendulum' for the inclined plane experiment); (ii) a *parameters* slot for the experimental parameters, represented as a list of *E!* frames (see §4.3.4); and (iii) an *exptsetups* slot containing experimental setups. The ease with which

Table 4.7 Experiment Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| exptparadigms | experimental paradigms | list of various expt paradigms frames |
| new_exptpardigms | rules for the invention of new experimental paradigms | *New paradigms* |
| perform_expttest | rules to design and perform experiments | *Experimenter* |

Table 4.8 Experimental Paradigm Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| name | experimental paradigm name | the name |
| parameters | experiment parameters | a list of E! frames |
| exptsetups | experimental setups | list of setups frames |
| manf_ease | measure of ease of manufacture | $0 \leq$ real number $\leq 1$ |
| bgknow_relate | relations for the interpretation of background knowledge | geometry & bodily entry frames |
| initparams | parameters of initial part of combined experiment | a list of E! frames |
| initials | initial expts. setups in combined experiments | experimental setups frames |
| terminals | terminal experimental setups in combined experiments | the experimental setups frames |
| bg_rel_initial | relations for interpretation of back ground knowledge, for initial experiments | the relations |

setups can be manufactured is specified as a number in the *manf_ease* slot (in the range [0 1]). The relation between sets of parameters and background knowledge is given by the special lists in the *bgknow_relate* slot (see §4.4.2).

The remaining four slots are only used when a combined experiment is invented and performed. There are slots for experimental setups that can act as the initial and terminal parts of the combined experiments, *initials* and *terminals*. The parameters for the initial part are stored in the *initparams* and their particular relations to the background knowledge are stored in the *bg_rel_initial* slot. The two equivalent slots for the terminal parts are the "normal" *parameters* and *bgknow_relate* ones. Combined experiments are named by the conjunctions of two ordinary names, for example 'incplane+projectile'.

*4.3.3.3 Experimental Setups*

The frame for experimental setups has four slots (see Table 4.9). The *name* slot holds labels such as 'down_incplane' and 'up_incplane' which are two of the setups for the inclined plane experimental paradigm. Parameters that are specific just to one setup are stored as a list of *E!* frames in the *parameters* slot. Experimental tests are contained as a list in the *expttest* slot and the *combine* slot contains the names of other setups that can act as initial parts if the present setup is the terminal part in a combined experiment.

*4.3.3.4 Experimental Test Representation*

The experimental tests frame has six slots that are equivalent to one in the instance frame on the theoretical side (Table 4.10). The *input-m*, *output* and *input–c* slots contain the manipulated input, measured output and the fixed input parameters, respectively, the last one being a list. The lists of values of the manipulated input and the measured output parameters are contained in the *input–m_vals* and the *output_vals* slots, respectively. The remaining slot, *terminal**, indicates the mode in which a combined experiment is used.

Table 4.9 Experimental Setup Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| name | experimental setup name | the name |
| parameters | parameters specific to setup | a list of E! frames |
| expttests | experimental tests | expttest frames list |
| combine | possible initial experimental setups for this setup | list of setup names |

Table 4.10 Experimental Test Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| input-c | fixed input parameters | list of E! frames |
| input-m | manipulated input (independent) parameter | an E! frame |
| input-m_vals | manipulated input parameter values | list of values |
| output | measure or observed output (independent) parameter values | an E! frame |
| output_vals | measure or observed output parameter values | list of values |
| terminal* | special experiment type indicator | symbol to specify terminal expt. |

### 4.3.4 Communication, Terms & Parameters

The last component of the framework deals with communication between the theory and experiment components. STERN models many types of communication between theory and experiment that typically involves the interrogation of fillers in particular slots in experimentation frames by theoretical inference processes, or vice versa. However, the correspondence between theoretical terms and experimental parameters requires specific representations.

*4.3.4.1 Correspondence By* Kinds *and* Qualkinds

The correspondence of specific theoretical terms and particular experimental parameters is a requirement by the framework, thus some means of mapping between them is required. To map between a term and its corresponding parameter, or *visa versa*, both are given an identical symbol. The symbol is called the *kind* of the term or the parameter in STERN. *Distance* and *time* are two examples of kinds needed to model the Galilean episode. However, previous work and the Galilean episode both demonstrate the importance of qualitative representations and inferences in discovery systems. Hence *qualkinds* are used for correspondence between terms and parameters that are merely observed but not measurable. For, example, the speed of an object can be qualitatively assessed even though it cannot be directly measured; *velocity* is qualkind of the speed term. Terms and parameters are in fact represented as frames with slots for the kind or qualkind.

Kinds and qualkinds in STERN are comparable to the units that exist in ABACUS (Falkenhainer & Michalski, 1986) and COPER (Kokar, 1986), although they have different roles in the different programs.

*4.3.4.2 T! term and E! parameter frames*

These are some of the most basic representations in STERN. However, STERN requires representations of terms and parameters with more structure than most previous discovery systems. The terms and parameters frames, called *T!* and *E!*,

have five and six slots, respectively (see Tables 4.11 and 4.12). Both have *kind* and *qualkind* correspondence symbols. STERN knows that a term and a parameter correctly match when contents of either the kind or qualkind slots are identical.

The *T!* frame has three other slots. The *val* slot contains the numerical value of the term when it is a directly measurable. Some quantitative terms can be defined as some function of other measurable terms, for example speed in terms of distance and time. The *equation* slot stores such definitions and the *nec_qualforms* slot contains qualforms (see §4.4.2 below) that specify the necessary conditions that must apply if the speed term is to be substituted for the distance and time terms (e.g., speed must be constant)..

The E! parameter frame also has a *val* slot for its magnitude, but in addition has *maxval* and *minval* slots defining the permitted ranges of the values given by the physical dimensions of the experimental setup. The *ease* slot contains a value (in the range [0 1]) that indicates the ease with which the parameter can be manipulated or observed within the experimental setup.

The extensive range of frames used to implement the components of the scientific framework in a general domain-independent manner have been described. To model a particular episode of discovery STERN needs domain specific knowledge, that is, the representations to fill the slots of the many frames just considered.

## 4.4 DOMAIN SPECIFIC KNOWLEDGE

We have considered knowledge representations that are supposed to be general to all episodes of scientific discovery. However, the character of scientific knowledge varies greatly from domain to domain. We now consider the representations that STERN possesses with which to model the Galilean domain.

Table 4.11 T! Theoretical Term Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| kind | quantitative kind of the term | eg. length, time |
| qualkind | qualitative qualkind of the term | eg. speed, density |
| equation | equation defining the term using other terms | an equation |
| nec_qualforms | necessary qualitative restrictions on substitutions using the equation | list of appropriate qualforms |
| val | value of the term | a real number |

Table 4.12 E! Experimental Parameter Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| kind | quantitative kind of the parameter | eg. length, time |
| qualkind | qualitative qualkind of the parameter | eg. speed, density |
| ease | a measure of the ease of manipulation and observation | $0 \leq$ real number $\leq 1$ |
| maxval | maximum value | real number |
| minval | minimum value | real number |
| val | current value | real number |

### 4.4.1 Quantitative and Qualitative Knowledge Representation

Galileo used a peculiar form of geometric-pictorial reasoning which would require a research project of its own to fully understand and model. Here, a more conventional approach has been adopted for both quantitative and qualitative representations.

First the quantitative representations are equations in theoretical terms. They terms may be related by simple arithmetic operators (i.e., products, quotients, summation, subtraction) and exponentials functions.The indices of the exponentials are small rational numbers - fractions with numerator and denominator as integers below a user specified limit, typically 3. Galileo expressed equations as ratios of terms, but here the equations that will be encountered will, for example, look like:

$$(= (* A B) (/ C D)). \qquad ...(4.2a)$$

In conventional mathematical notation is written as:

$$A . B = C / D . \qquad ...(4.2b)$$

The equality sign is not however to be read conventionally but means proportionality, thus making (4.2a) equivalent to the Galilean expressions using ratios of terms.

The second form of domain specific knowledge are qualitative relations that Galileo expressed propositionally. Here, such relations are stated in the form of Qualitative Formalisms, called *qualforms*. They employ a predicate argument like notation; for example:

$$(\text{increase } A\ B), \qquad ...(4.3a),$$

and

$$(\text{decrease } C\ D), \qquad ...(4.3b).$$

where A, B, C and D are theoretical terms. The two qualforms assert, respectively, that "A increases as B increases" and "C decreases whilst D increases". Table 4.13 lists all the qualforms used in STERN. Although qualforms bear a resemblance previous approaches to qualitative reasoning, such as IDS (Nordhausen & Langley,

Table 4.13 Qualforms and their interpretations

| Qualform | Variation of A as B increases uniformly |
|---|---|
| (INCREASE A B) | A increase |
| (DECREASE A B) | A decreases |
| (STEADY A B) | A is constant |
| (LINEAR A B) | A uniform rate of increases from zero |
| (INSTANTANEOUS A B) | A is constant after brief initial rapid increase |
| (REPEAT+ A B) | A rises to a maximum and returns to its starting value |
| (REPEAT- A B) | A falls to a minimum and returns to its starting value |
| (INDEPENDENT A B) | None of the above. |
| (FROM_ZERO A B) | 1st A and B values at origin |

Table 4.14 Background Knowledge ENTRY Frame Slots and Fillers

| Slot name | Description | Filler in STERN |
|---|---|---|
| variable | 3 variables in any order | eg. arc_x, arc_l, arc_@ |
| fixed | the variable held constant | eg. arc_@ |
| equation | quantitative relation between all 3 variables | eg. (= arc_x (* arc_l arc_@)) |
| qualform | qualitative relation between the 2 unfixed variables if no equation | eg. (increase arc_x arc_l ) |
| limits | pairs of boundary values or substitutible variables | eg. '((0 0) (pi/2 arc_d)) |

1987) the role they play in STERN is quite different. Qualforms are also similar to BACON like regularity spotters (Langley *et.al.*, 1987; Falkenhainer & Michalski, 1986; Reimann, 1990) but qualforms explicitly state the relation noted in a manipulable form and cover a much wider range of relations. Qualforms are central to STERN's integrated quantitative and qualitative discovery abilities.

### 4.4.2 Background Knowledge

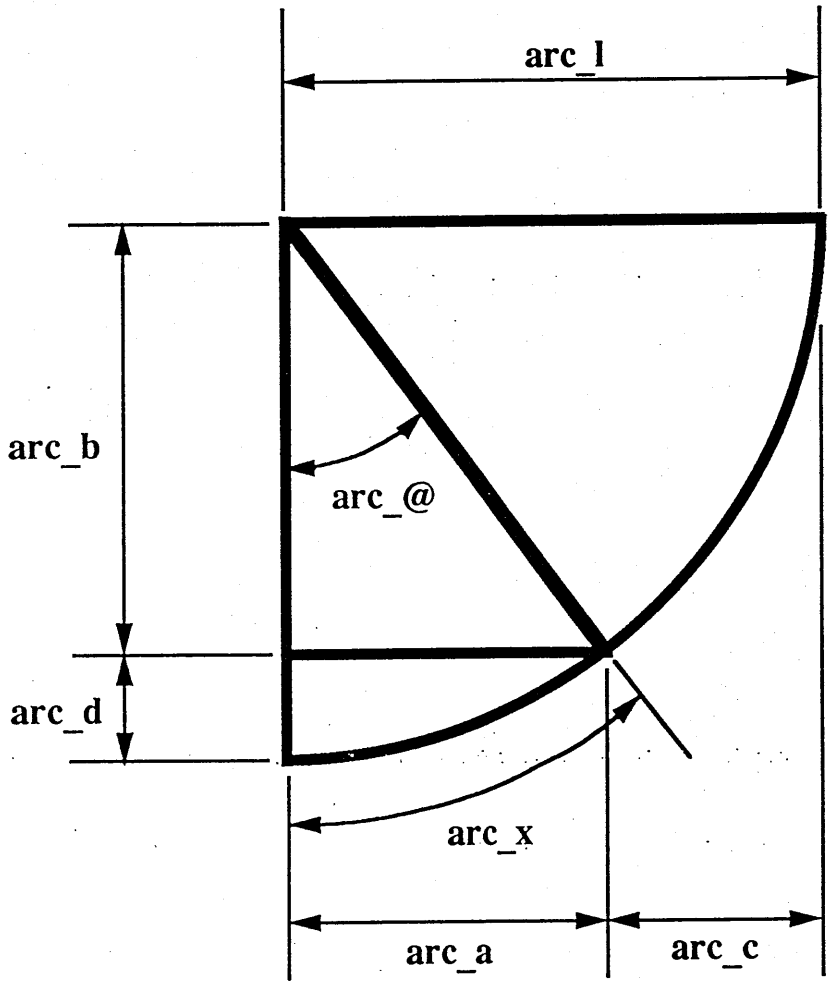
*4.4.2.1 Entry Frames To Store Relations*

An example of background knowledge in the Galilean domain is the understanding of geometry that underlies the geometric-pictorial form of reasoning. In various different types of inference STERN also employs a repository of background geometrical knowledge that summarizes basic trigonometrical relations. Figure 4.2 shows the geometric relations. To store this and all other types of background knowledge, STERN uses a list of *entry* frames, headed by a symbol (eg. arc) that names the list. Each frame has five slots that defines the relation between certain variables (Table 4.14). The *variables* slot contains a list of three variables that are interrelated by the *equation* or *qualform* stored in slots with those names, where the variable named in the *fixed* slot is held constant. The ranges over which the other two variables can vary are specified in the *limits* slot. Two of STERN's eleven geometric, or arc, entry frames are included in Figure 4.2. STERN also possesses background knowledge for bodily relations of spherical objects.

*4.4.2.2 Background Knowledge Mapping Relations*

To be able to use background knowledge in the form just described the theory and the experimental paradigm frames have *bgknow_relate* (and *bg_rel_initial*) slots. These slots contain lists that define the relation between the theoretical or experimental knowledge and background knowledge entry frames. The lists map terms onto the background knowledge variables. Each list starts with the name of a set of background knowledge (e.g. arc) and is followed by pairs of terms and

**Relation between the variables**




**Two example definitions of entry frames**

```
(list
  'arc
  (make-entry :variables '(arc_x arc_l arc_@)
              :fixed 'arc_@
              :equation '(= arc_x (* arc_l arc_@)))
  (make-entry :variables '(arc_c arc_l arc_@)
              :fixed 'arc_l
              :qualform '(decrease arc_@ arc_c)
              :limits '((0 arc_l) (*pi/2* 0)))
   . . . . . .
)
```

**Figure 4.2 Geometric Background Knowledge**

background variables. STERN uses the lists to transform theoretical terms into background variables. The variables are then used to find suitable entry frames and the information in the frames is used to make inferences. For example, a qualform from the entry frame may be transformed into an equation using the specified limits. The equation is then transformed back into theoretical terms using the mapping lists in reverse.

Two examples of the way background knowledge is used in STERN are: (i) the elimination of non-measurable terms by substitution and (ii) the identification of terms or parameters that are trivially related due to the physical geometry of an experimental setup rather than via the phenomenon.

### 4.4.3 Summary

We have considered all of STERN's knowledge representations. A clear distinction has been made between the general representations to implement the present framework and domain-specific representations to model the Galilean domain. The framework representations have been designed so that it should be possible, in principle, to use an alternative collection of domain specific representations without altering the general frames. This distinction based on knowledge specificity has been maintained in the design of STERN's processes and rules. We will now consider STERN's processes and rules.

## 4.5 DISCOVERY PROCESSES AND RULES

We have seen STERN's many different knowledge representations. Here, we will consider the processes used by STERN to manipulate this knowledge in order to make discoveries. We will see how the the framework has been used to define specific classes of processes to perform particular tasks and how STERN instantiates the processes as rules. A production system-like architecture is used for overall programmatic control.

### 4.5.1 Knowledge States And Condition-Action Rules

We have seen how the framework provides a principled organization of theoretical and experimental knowledge in STERN. The framework has also been used to organize STERN's discovery processes.

The framework posits that three types of theoretical knowledge and three levels of experimentation are present in a research programme. Each has associated information (e.g., measures of acceptability) and may be under active consideration at a particular time or simply held in long-term memory. Thus it is possible to identify different *states* of a research programme in terms of: (i) the presence or absence of particular items of knowledge; (ii) whether those items are active; and (iii) the magnitudes of the various types of information related to each item. This state based characterization of research programmes is used in two different ways in the specification of rules in STERN.

First, the states of a research programme are used to express STERN's general domain-independent discovery processes as well defined condition-action rules. A particular state defines the condition and the action specifies some definite change to be made to the state. For example, a condition may find that there is an active acceptable hypothesis, but that the hypothesis does not account for a given experimental setup (i.e., the hypothesis does not possess any relevant models). The action of the rule would then use the hypothesis to generate models for the experimental setup.

Second, *tasks* are defined in STERN in terms of particular states and general changes to those states, rather like rules. However, tasks differ from rules in that they are comprised of many rules. For example, the testing of a model employs several rules including those to: generate predictive instances; to design and perform experiments for each instance; and to compare each experimental result with an instance. Tasks may engage sub-tasks to carry out specific procedures; generating

predictive instances is a sub-task of the above model testing example. We will see below, all of STERN's rules are classified into groups in a well defined task hierarchy. Tasks in STERN are similar to *partitioned* or *packaged* rules in previous discovery programs (e.g., BACON, Langely *et.al.*, 1987; KEKEDA, Kulkarni & Simon, 1988). However, the state based approach used here is more principled. The context of each group of rules in STERN within the overall discovery process is transparent. Furthermore, the grouping of rules according to a particular task means that STERN possesses a property that is very desirable in discovery systems; namely, the unambiguous separation of domain-specific and domain-independent processes. In STERN the classes of rules that are applicable to a wide variety of episodes of discovery are clearly distinguished from those that apply only to the Galilean domain. Contrast this with KEKEDA (Kulkarni & SImon, 1988) in which domain-specific and domain-independent heuristics are found in every group of rules.

To summarize, the framework provides a principled method by which to define rules and to organize them into groups within a clearly defined hierarchy of tasks. We now consider the groups of rules in themselves and the tasks they perform.

### 4.5.2 STERN's Classes Of Rules

STERN has 64 rules. They are grouped into 16 classes that perform specific tasks. There are domain-independent and domain specific sets of tasks (see Tables 4.15 & 4.16, and Figure 4.3).

The domain-independent rules are built into STERN. They control the way STERN makes discoveries; guiding it down different paths as the process evolves. These rules typically refer to frames that instantiate the components of the framework without changing the contents of the slots.

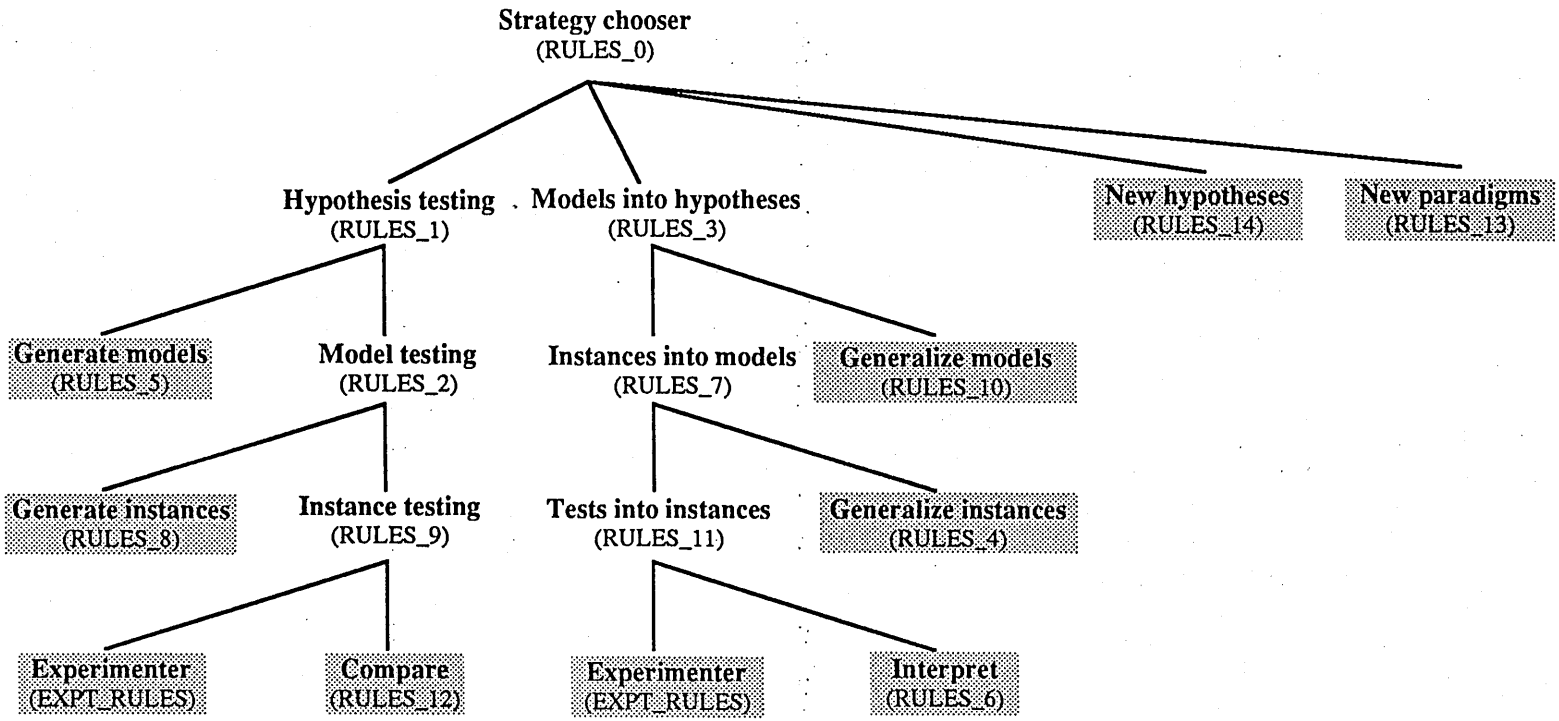
The domain specific rules, on the other hand, tell STERN how to perform particular inferences on the Galilean knowledge representation; such as equations and qualforms. The groups comprise the procedural knowledge for the Galilean

Table 4.15 General Domain Independent Classes Of Rules

| Name (STERN reference) | Description |
|---|---|
| Strategy chooser (RULES_0) | Top level control and strategy chooser |
| Hypothesis testing (RULES_1)<br>Model testing (RULES_2)<br>Instance testing (RULES_9) | Hypothesis testing by experimental paradigm selection and model generation<br>Model testing by experimental setup selection and instance generation.<br>Instance testing by experimental test design, performing the tests, and results comparison. |
| Models into hypotheses (RULES_3)<br>Instances into models (RULES_7)<br>Tests into instances (RULES_11) | Selection of experimental paradigms to obtain models to be generalised into hypotheses.<br>Selection of experimental setups to obtain instances to generalise in to models.<br>Design and performance of experimental tests, to permit the interpretation of experimental results into instances. |

Table 4.16 Domain Specific Classes Of Rules

| Name (STERN reference) | Description | Location in STERN (frame: slot) |
|---|---|---|
| Generate models (RULES_5)<br>Generate instances (RULES_8)<br>Compare (RULES_12) | Qualitative and quantitative model generation from hypotheses & experimental paradigms<br>Qualitative & quantitative instance generation from models & experimental setups<br>Comparison of qualitative and quantitative instances and experimental tests. | Theory: Generate_models<br>Theory: Generate_instances<br>Theory: Instance_vs_expttest |
| Generalize models (RULES_10)<br>Generalize instances (RULES_4)<br>Interpret (RULES_6) | Generalization of qualitative and quantitative models into hypotheses.<br>Generalization of qualitative and quantitative instances into models.<br>Interpretation of experimental test results into instances. | Theory: Generalise_models<br>Theory: Generalise_instances<br>Theory: Interpret_expttests |
| New paradigms (RULES_13) | Invention of new combined experimental paradigms. | Experiment: New_exptpardigms |
| New hypotheses (RULES_14) | The generation of new qualitative and quantitative heuristics form existing ones. | Theory: Generate_hypos |
| Experimenter (EXPT_RULES) | (i) Design of experimental tests given a setup and an instance.<br>(ii) Experiment simulator. | Experiment: Perform_expttest |




## Figure 4.3 Hierarchy Of STERN'S Classes Of Rules

Shaded names are domain specific classes of rules.
(STERN reference in parentheses)
See Tables 4.14 & 4.15 for descrptions of the classes.

research programme. STERN knows which group to invoke by referring to slots in the theory and experiment frames (see Table 4.16). For example, when new hypotheses need to be generated from old, the rules named in the *generate_hypos* slot of the theory frame are retrieved (i.e., the *new hypotheses* class). STERN is run by invoking the *strategy chooser* (RULES_0[2]) class of rules. These rules govern STERN's overall performance by selecting one of four very different high-level tasks. These tasks are the four general discovery processes that were identified in the Galilean episode (see Chapter 3): (i) the confirmation of existing of hypotheses - *hypothesis testing* (RULES_1); (ii) the generalization from experimental results into new hypotheses - *models into hypotheses* (RULES_3); (iii) the formation of new hypotheses from existing ones - *new paradigms* (RULES_13); and (iv) the invention of new experiments - *new hypotheses* (RULES_14).

In STERN, confirming a hypotheses depends on the fact that the acceptability of hypotheses is a function of the acceptability of models, and that in turn, the acceptability of models is a function of the acceptability of instances. The process starts with the *hypothesis testing* class of rules (see Figure 4.3). These rules generate models to account for a particular experimental paradigm and assesses the acceptability of the hypotheses with respect to each model. Models are formed by the domain specific *generate models* rules. Each model is individually investigated by the *model testing* rules. Testing a model has a similar pattern to assessing a hypothesis; specifically, instances are generated with respect to an experimental setup and the acceptability of the models is assessed according to the success of the instances. Instances are formed by *generate instances*. Individual instances are examined by *instance testing*. Testing an instance involves the design and performance of an experimental test, by the *experimenter* rules, and the comparison

[2]Each class of rules has a reference name (e.g., RULES_0) so that the class of an individual rule can be simply identified by its 'R*x*_' prefix, where *x* is the reference number of the class (e.g., R0_START_CONFIRM).

of the result with the instance, by the *compare* rules. We will see in detail how STERN performs confirmation in Chapter 5.

STERN forms hypotheses from experimental results by generalizing models. Models are obtained by generalizing instances, and instances are interpreted experimental test results. *Models into hypotheses* has the job of finding the models (Figure 4.3), using *instances into models*. The models are generalized to form hypotheses by *generalize models*. Models are obtained by *instances into models* that finds instances, using *tests into instances*, and forms them into models, using *generalize instances*. In turn, instances are found by performing experiments, *experimenter*, and interpreting the test results in to instances, *interpret*. We will see in detail how STERN obtains hypotheses in Chapter 6.

The generation of new hypotheses from old only involves one set of rules, *new hypotheses* (RULES_14). STERN only examines the one type of theoretical knowledge when carrying out this task and thus does not need multiple sub-tasks. The generation of new hypotheses from old is considered in more detail in Chapter 7.

Inventing new experimental paradigms also requires just one set of rules, *new paradigms* (RULES_13). Again, since only one type of experimental knowledge is considered, the need for additional classes of rules is obviated. The invention of new experimental paradigms is considered in more detail in Chapter 8.

Although the organization of rules given in Figure 4.3 follows from the framework it is not the only possibility. Thus the hierarchy of rules that STERN employs can be considered as a "hypothesis" that will be assessed according to the acceptability STERN as a model of discovery.

In the review of previous work (in Chapter 2) it was concluded that it is desirable for discovery systems to (i) possess a richness of heuristics to match KEKEDA (Kulkarni & Simon, 1988) but (ii) organized in a principled manner

rather like the SDDS process hierarchy (Klahr & Dunbar, 1988). STERN's 64 rules in 16 groups, organized in the structure given in Figure 4.3, satisfy this dual requirement.

### 4.5.3 Production System Implementation

STERN's processes are represented as condition-action rules implemented in a production system like architecture. The system is loosely based on Hasemer & Domingue's (1989) production system, but has substantial alterations. In STERN, the working memory is the complex structure of frames described above plus a *current* frame. This frame has slots for the three types theory and the three levels of experiments. The slots (*hypothesis*, *model*, *instance*, *exptparadigm*, *exptsetup* *expttest*) are filled by hypotheses, models, etc. that are being actively considered at a particular time. The rules themselves are represented as *rule* frames with *antecedent* and *consequent* slots for the conditions and actions, respectively. Each rule has a *name* slot and a *counter* slot that is incremented each time the rule is fired. The conditions and action are CL functions. The condition can examine, but never change, the contents of slots of any of the systems knowledge structures. The actions, however, have full access to knowledge structures; they can manipulate the stored information and instantiate new items in the structure. (Table 4.17 gives the CL code for one rule.[3])

Like more classical productions systems STERN has cycles that involve: (i) the matching of rules against working memory; (ii) conflict resolution to select one rule to fire from amongst those that are successful in a cycle; (iii) firing the rule and the looping back to the matching stage. The conflict resolution strategy used by STERN normally chooses the rule with highest priority not fired in the last cycle. The order of the rules in a class determines their priority. In certain cases the not-fired-in-the-last-cycle condition is suspended as some classes of rules iterate over a list of items (i.e. *strategy chooser* and *generate models*).

[3]A disk with all the CL code for STERN is included with this thesis (or may be obtained on application to HCRL).

## Table 4.17 CL Code For R1_HYPO_ASSESS_WRT_MODELS Rule.

```
;;;-------------------------------------------------------------------------
;;; R1_HYPO_ASSESS_WRT_MODELS_ANT
;;; Function to instantiate rule antecedent for assessment
;;;          of an hypothesis with respect to a model
;;; Condition: there is a current hypo & a current model that has been tested
;;; Version 1.2  29/3/90   (V1  20/9/89)
;;; Returns: T if Condition satisfied, nil otherwise
;;; Calls:  untested?  current  c_slot
;;; Arguments: --
;;; Variables: --
;;; Structures: current hypo model
;;;
(defun  R1_hypo_assess_wrt_models_ant ()
              ;;an active hypothesis
  (cond ((and (current hypo)
              ;;an active model
              (current model)
                  ;;the model has been successfully tested
              (or (null (untested? (c_slot model adequacy)))
                  ;;unsuccessful attempts have been made to test the model
                  (c_slot model not_tested))))))

;;; R1_HYPO_ASSESS_WRT_MODELS_CON
;;; Function to instantiate rule consequent of hypo assessment
;;;          with respect to a model.
;;; Version 1.1  13/3/90    (V1  20/9/89)
;;; Action: Assesses the hypothesis acceptability, deactivates the model,
;;;         and clears expt paradigm if all models have been tested.
;;; Calls: statement  clear_current  current  amend_measure
;;; Arguments: --
;;; Variables: --
;;; Structures: hypo  model  measure
;;;
(defun R1_hypo_assess_wrt_models_con ()
  (setf (c_slot hypo adequacy)
        (amend_measure (c_slot hypo adequacy)
                       (/ (measure-degree (c_slot model adequacy))
                          (measure-number (c_slot model adequacy)))))
  (clear_current model)
  (statement 'rules_1 "hypo adequacy =" (c_slot hypo adequacy)
             "and current model cleared")
  ;;clear current paradigm only if all models have now been tested
  (cond ((null (untested_hypo_models (c_slot hypo models)))
         (clear_current exptparadigm)
         (statement 'rules_1 "All models tested, exptparadigm cleared"))))

;;;-------------------------------------------------------------------------
```

**Notes:**

All frames are instantiated as `defstructs`

`(c_slot` *`frame slot`*`)` - retrieves the filler of *slot* in the active *frame*

`(untested?` *`measure`*`)` - checks whether the item with the specified *measure* frame has been tested

`(statement` *`items`*`)` - prints the *items* to the output stream

`(clear_current` *`frame`*`)` - makes the *frame* inactive

`(amend_measure` *`measure amount`*`)` - increments the *measure* frame by the given amount.

In conventional production systems with large numbers of rules, the rules are often packaged into groups with common tests in their conditions. However, STERN's sixteen sets of rules need to be organized hierarchically as shown in Figure 4.3, so a different technique is employed. This technique involves recursive calls to the PS with particular class of rules. A new class of rules (on level $\hat{m}$-*1* of the hierarchy) can be invoked from a single rule (on level *m*) by recursively calling the PS with the new rules. For example, the action of the R0_START_CONFIRM rule, in the *strategy chooser* class, calls the PS with the *hypothesis testing* rules. The PS to cycle through the *hypothesis testing* rules until, eventually, none match. Control is then popped back up one level to the PS with the *strategy chooser* rules. Two of the *hypothesis testing* rules make recursive calls (to *generate models* and *model testing*). The whole hierarchical structure of rules is implemented using this simple technique.

## 4.6 EXPERIMENT SIMULATION

Unlike previous systems, where the user is responsible for supplying hand calculated experimental results, STERN possesses an experiment simulator. All but one of the *experimenter* (EXPT_RULES) class of rules comprise the subprogram that simulates the performance of experiments (the exception designs experimental tests). This has no bearing of STERN's discovery abilities but obviates the need for user to supply experimental results as inputs. STERN has no access to the contents of the simulator other than via its inputs and outputs. The simulator is given specified experimental parameters as inputs and calculates the values of the parameters as its outputs.

### 4.6.1 Black Box Simulator

The black box conceptualization of experiments given by the framework suggests a technique for the simulation of the phenomenon in an experiment.

Generally, in science the objective is to discover the contents of the "black box" phenomenon that functionally relates the input and output parameters. However, for the Galilean motion domain, a high level characterization of the phenomena already exists; namely the conservation of energy, as expressed as the constant magnitude of the sum of potential (height) and kinetic (speed) energies, throughout the movement of a body. Thus, given an input parameter and its successive values, it is possible to determine the corresponding output parameter values, by calculating the transfer of potential to kinetic energy .This is a feasible proposition because the all the experimental paradigms in the domain have parameters that are related to height and speed by their physical geometry. This forms the conceptual basis of STERN's rules that simulates the performance of real experiments.

### 4.6.2 Implementation And Performance

The *experimenter* is the set of rules instantiating experimental performance. All but one simulates the performance of experiments under particular experimental setups. (The exception is used to prepare experimental tests when an instance is available. It will be described in Chapter 5.) The rules are instantiated by such a central function, that incrementally varies the input parameter in order to calculate the series of output values. However, the rule to perform combined experiments is different, because separate simulations are required for the initial and terminal parts. However, the same principle is used for each of the parts in turn, and then the two are combined.

Noise is artificially added to the calculated output values. Each value is altered by a random amount within a band given by a certain specified percentage of the calculated value. The new noisy value can be any where within the band, with an approximately uniform probability. Settle's (1961) experiments show that the noise level given by a half band width of 2 percent is realistic for the Galilean domain.

We have just seen how STERN simulates experiments. Earlier we considered STERN's knowledge representations and classes of rules. Now we can begin to

examine how STERN brings all these pieces together to make discoveries. The next section describes how STERN chooses which of the four main discovery processes to carry out.

## 4.7 TOP LEVEL CONTROL - *Strategy Chooser*

STERN has four main discoveries making processes (i.e., confirming hypotheses, generalization from experimental results, generating new hypotheses from old, and the invention of new experiments). Choosing which to carry out at any particular time is the task of the *strategy chooser* rules. They constitute STERN's top level of global control. Invoking the production system with *strategy chooser* runs the program.

Four rules comprise the *strategy chooser* class (see Table 4.18). The priority of the rules in production system conflict resolution is the same as their order in Table 4.18. The action of all four rules is to make recursive calls to the production system with sets of rules to carry out the particular strategy chosen. STERN chooses the confirmation strategy (using R0_START_CONFIRM) when there are hypotheses that have not been used to account for all experimental paradigms, successfully or otherwise. For example, given the *effective weight law* (hypothesis) as an input, STERN tries to confirm the hypothesis by applying it to the pendulum and inclined plane experiments. The generalization of experimental results into hypotheses (R0_START_INDUCE) occurs when experimental paradigms have not been successfully accounted for by a hypothesis. For example, STERN performs experiments on the pendulum experiments to find equations and qualforms for generalization. New hypotheses are inferred from old (R0_NEW_HYPOS) when attempts to (dis)confirm all existing hypotheses have been made and hypotheses have been generalized from experimental results. The law of free fall is found by STERN using this strategy. New experimental paradigms are considered when hypotheses have not been (dis)confirmed but all existing paradigms have been tried

## Table 4.18 Strategy Chooser Rules (RULES_0)

**R0_START_CONFIRM***
**Condition:**
There are stored hypotheses that are completely untested or have not been tested against all manufactured experimental paradigms.
**Action:**
Call PS with *Hypothesis testing* (RULES_1) to examine the untested hypotheses in conjunction with the experimental paradigms with which they have not yet been tested by generating models.

**R0_START_INDUCE**
**Condition:**
All hypotheses have been examined irrespective whether they were tractable,
and not all the manufactured experimental paradigms have been successfully accounted for by a hypothesis.
**Action:**
Call PS with *Models into hypotheses* (RULES_3) to perform experiments on the unaccounted for experimental paradigms in an attempt to generalization the results into hypotheses via models.

**R0_NEW_HYPOS**
**Condition:**
All stored hypotheses have been tested using at least one model
**Action:**
Call PS with the set of rules in the theory frame generate_hypos slot (i.e., *New Hypotheses*, RULES_14) to generated new hypotheses from existing ones.

**R0_NEW_ EXPTPARADIGMS**
**Condition:**
There are stored hypotheses that remain untested.
**Action:**
Call PS with rules in the experiment frame new_experiments slot, (i.e., *New paradigms*, RULES_13) to make new combined experiments

*The order of rules indicates their relative priority in conflict resolution.

(R0_NEW_EXPTPARADIGMS). For example, STERN needs the combined inclined plane and projectile experiment as the law of free fall cannot be tested using the pendulum or inclined plane paradigms.

We now know the circumstances in which STERN chooses to employ each strategy. The four main strategies will be considered in their own chapters, 5 to 8.

Chapter 5 - The confirmation or otherwise of existing hypotheses;

Chapter 6 - The performance of experiments to obtain results that are generalized into hypotheses via models;

Chapter 7 - The generation of new hypotheses from a stock of existing hypotheses with varying degrees of acceptability; and,

Chapter 8 - The consideration of new experiments and the pragmatic selection of paradigms.

The chapters will also compare STERN's abilities and those of existing discovery systems, with particular attention to the manner in which STERN attempts to overcome their limitations.

# Chapter 5

# Confirming Existing Hypotheses

## 5.1 INTRODUCTION

At the beginning of his investigations Galileo adopted the Aristotelian laws current in his day. He attempted to demonstrate these laws were correct by making predictions about the behaviour of balls rolling down inclined planes and the motion of swinging pendulums. Through successive experimental trials, Galileo tested these predictions but found they were poor. Much later in the episode Galileo inferred his law of free fall. The acceptability of this law was determined by the making of predictions for comparison with experimental results. These two examples show that predictive (dis)confirmations of known hypotheses plays an important part in scientific discovery.

STERN models this important aspect of discovery. On four occasions during STERN's run (see Chapter 4) known hypotheses came to be tested against experimental outcomes. Each time this happened STERN acted in a theory-led fashion and invoked the confirmation strategy (rule R0_START_CONFIRM in *Strategy chooser*).

This chapter considers the details of how STERN models the process of predictive confirmation of existing hypotheses. After an overview of the process is given, I will present a detailed account of the stages in the process. Finally, comparisons to previous work will be made.

## 5.2 STAGES IN CONFIRMING A HYPOTHESIS

Confirming a known hypotheses requires many different classes of rules invoked on many occasions to perform specific tasks. However, the whole process

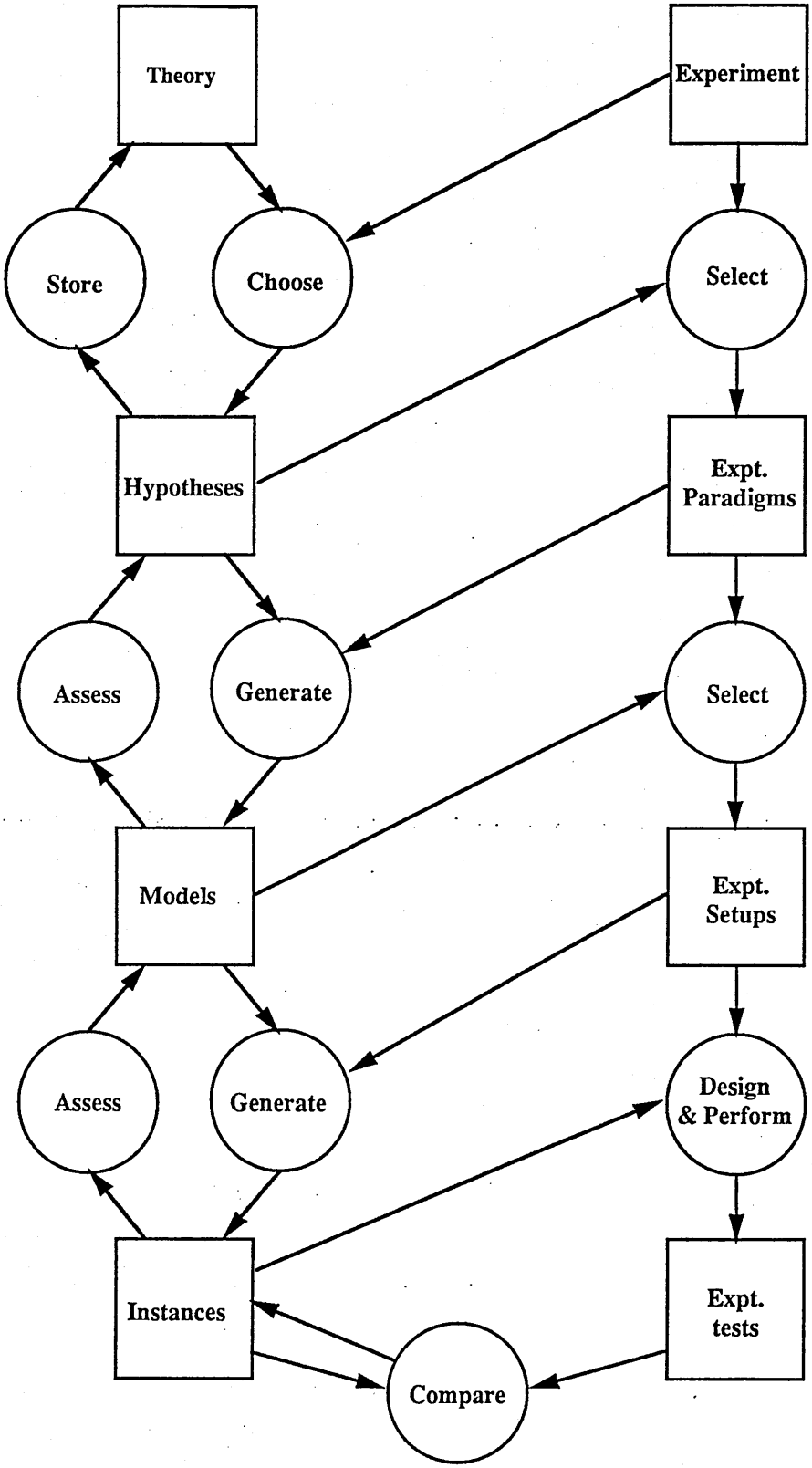



Figure 5.1 STERN's Confirmation Strategy

## Table 5.1 HYPOTHESIS_TESTING Rules (RULES_1)

**R1_CHOOSE_MODEL***

**Condition:**

There is no active model, but an active hypothesis and experimental paradigm,
and the active hypothesis has 1 or more models and no attempts have been made to test them.

**Action:**

Make active the first stored model on which no attempts at testing have been made.

**R1_HYPO->MODELS**

**Condition:**

There is no active model, but an active hypothesis and an active experimental paradigm,
and there are no stored hypotheses with models that account for the active experimental paradigm.

**Action:**

Call the PS with the domain specific rules whose name is in the generate_models slot of the theory frame (i.e.,*Generate models*, RULES_5) to infer models from the active hypothesis and paradigm.

On return amend tractability, and if no models have been generated deactivate the experimental paradigm.

**R1_CHOOSE_PARADIGM_WITH_HYPO**

**Condition:**

There is an active hypothesis, but no active experimental paradigm,
and there are stored manufactured experimental paradigms that have not been tested with the active hypothesis.

**Action:**

Make active the manufactured experimental paradigm with the greatest product of number of setups and ease of manufacture.

**R1_HYPO_ASSESS**

**Condition:**

There are no active models,
and there is an active hypothesis that has been tested with all manufactured paradigms,
and attempts have been made to test all stored models of the active hypothesis.

**Action:**

Store the hypothesis (ie. make it inactive),
and if it is acceptable and a member of a group, then make the other members unacceptable.

**R1_HYPO_ASSESS_WRT_MODELS**

**Condition:**

There is an active hypothesis and an active model,
and attempts have been made to test the model.

**Action:**

Calculate the acceptability of the model using equation 4.1 and use it to amend the acceptability measure of the hypotheses,

And make the model and the experimental paradigm inactive.

*The order of rules indicates their relative priority in conflict resolution.

continued . . .

## Table 5.1 Continued HYPOTHESIS TESTING Rules

**R1_TEST_MODEL**
**Condition:**
There is an active model and no attempts have been made to test it.
**Action:**
Call the PS with *Model testing* (RULES_2) to test the model by instance generation.

**R1_CHOOSE_HYPO**
**Condition:**
There is no active hypothesis,
and there are stored hypotheses that are have not been or only partly tested against all the manufactured experimental paradigms.
**Action:**
Make active a stored hypothesis that has not already been fully tested preferring quantitative hypotheses.

**R1_CHOOSE_PARADIGM_NO_HYPO**
**Condition:**
There is an active hypothesis, but no experimental paradigm,
and some paradigms have not already been tried with the hypothesis.
**Action:**
Make active the manufactured experimental paradigm with the greatest product of number of setups and ease of manufacture.

can been be summarized diagrammatically, see Figure 5.1. The boxes are levels of knowledge in the framework, and the circles are inference processes. STERN uses the *hypothesis testing* rules (Table 5.1) to control the confirmation process.

STERN begins the assessment of the acceptability of a hypotheses by choosing an experimental paradigm (R1_CHOOSE_PARADIGM_WITH_HYPO) and a hypothesis (R1_CHOOSE_HYPO). The hypothesis is not one that has previously been tested with all the stored experimental paradigms. Models are then generated using not only the hypothesis but also knowledge about the experimental paradigm (R1_HYPO->MODELS). STERN chooses one model to test (R1_CHOOSE_MODEL & R1_TEST_MODEL). This involves selecting an experimental setup with which the model is used to generate predictive instances. For each instance an appropriate experimental test is designed and performed. The instance and experimental test results are then compared; this involves STERN calculating the degree of predictive accuracy of the instance. This value is used to determine the acceptability of the model and the acceptability of the model is used to assess the acceptability of the hypothesis (R1_HYPO_ASSESS_WRT_MODELS). When a hypotheses has been tested against all experimental paradigms it is stored and another hypothesis chosen (R1_HYPO_ASSESS).

STERN may generate several instances from one model. Thus, STERN employs repeated cycles of the processes to test a model using each instance (i.e. the *Instance testing* rules, Table 5.2). The cycles are comprised of (i) the design and performance of an experimental test, (ii) comparing the test results and the instance, and (iii) assessing the model given the degree of success of the instance. Similarly, several models may be generated from one hypotheses. Thus, there are cycles of processes to test each hypothesis using the models (i.e. the *Model testing* rules, Table 5.3). These cycles include: (i) selecting an experimental setup; (ii) generating the instances from the model; (iii) testing all the instances to assess the

## Table 5.2 INSTANCE TESTING Rules (RULES_9)

**R9_TEST_INSTANCE***
**Condition:**
There is an active instance,
and an active experimental test with a fully specified set of results.
**Action:**
Call PS with rules named in the instance_vs_expttest slot of the theory frame (i.e., *Compare*, RULES_12) to compare the instance and experimental test,
and deactivate the experimental test.

**R9_NO_TEST_PERFORMED**
**Condition:**
There is an active instance,
and an active experimental test but no experiment has been successfully performed.
**Action:**
Make the experimental test inactive,
and set the instance acceptability (degree slot) to -1 as flag noting that no experiment was performed.

**R9_PERFORM_EXPT_TEST**
**Condition:**
There is an active instance that has not been compared with an experimental test,
and an active model and experimental setup,
and no active experimental test.
**Action:**
Call PS with the set of rules named in perform_expttest of the experiment frame (i.e., *Experimenter*, EXPT_RULES) to design and perform an experiment to match the active instance using the active setup.

*The order of rules indicates their relative priority in conflict resolution.

## Table 5.3 MODEL TESTING Rules (RULES_2)

**R2_NO_MORE_INSTANCES***
**Condition:**
There is an active experimental setup, and no active instance,
and an active model with no stored instance on which attempts have been to test with the active setup.
**Action:**
Make the current experimental setup inactivate

**R2_ASSESS_MODEL**
**Condition:**
There is an active model,
and an active instance on which attempts at comparison with experimental tests have been made.
**Action:**
Amend the acceptability of the model paying attention to non comparison situations because of failures to perform an experimental test.

**R2_TEST_INSTANCE**
**Condition:**
There is an active instance and no attempt has been made to compare it with an experimental tests.
**Action:**
Call PS with *Instance testing* (RULES_9) to test the instance.

**R2_CHOOSE_INSTANCE**
**Condition:**
There is an active experimental test,
and an active model with associated instances,
and no active instance.
**Action:**
Make active the first instance stored with model.

**R2_CHOOSE_SETUP**
**Condition:**
There is no active experimental setup, and an active model,
and stored setups that have no been tried with the model.
**Action:**
The first untried experimental setup is made active.

**R2_OBTAIN_INSTANCES**
**Condition:**
There is an active experimental setup and an active model,
and no instance associated with the model,
and the setup has not been considered before by the model.
**Action:**
Call the PS with the named rules in the generate_instance slot in the theory frame (i.e., *Generate instances*, RULES_8) to make instances; if successful amend the model's tractability and store the instances under the model, otherwise deactivate the experimental setup.

*The order of rules indicates their relative priority in conflict resolution.

model (i.e. sets of the cycle just described); and (iv) assessing the hypothesis with respect to the model. Hence, the confirmation of hypotheses by STERN can be viewed as a process with nested cycles of sub-processes (see also Figure 4.2).

However, STERN's confirmation strategy can be broken down in to three distinct stages. We can think of the process as being comprised of: (i) making predictive instances from a hypothesis; (ii) comparing the predictions with experimental tests; and (iii) assessing the acceptability of the hypothesis using the instances. We will now consider the details of confirmation strategy using these three stages. Seven classes of rules are involved in the confirmation process (see Tables 5.1 to 5.7). The priority of the rules in each class are given by their order (in each table).

## 5.3 MAKING PREDICTIONS

STERN begins the confirmation processes by choosing the pendulum experiment, from amongst those that have been made *available* (see Chapter 8). It is selected as it is the most easily manufactured and has the most setups. The first hypothesis to be (dis)confirmed by STERN is one of the effective weight hypotheses, with the equation:

$$(= \text{T\_V T\_DEN}) , \qquad \ldots (5.1)$$

where T_V and T_DEN are *speed* and *density* terms. This hypothesis is chosen because it has not been considered before with the pendulum paradigm. STERN can now attempt to generate models from hypothesis for the chosen experimental paradigm.

### 5.3.1 Generating Models

STERN generates models from quantitative and qualitative hypotheses (using the *Generate models* rules, Table 5.4).

#### *5.3.1.1 Typical Quantitative Model Generation*

The effective weight law, Equations 5.1, is a typical form hypothesis that

## Table 5.4 GENERATE MODELS Rules (RULES_5)

**R5_COMB_COMPLEX_EQN†**
**Condition:**
There is an active quantitative hypothesis and all of its associated models have been tested,
and there is an active paradigm that is a combined experiment,
and the equation does not have terms that are all measurable.
**Action:**
Attempt to generate models for the combined experiment in the initial and terminal modes in turn, by trying to find a substitute for one theoretical term that is not measurable by examining suitable background knowledge,
and amend the tractability of the hypothesis according to the success of the action.

**R5_COMPLEX_QUALFORM**
**Condition:**
There is an active qualitative hypothesis and all of its associated models have been tested,
and at least one of the terms of the qualform is not measurable.
**Action:**
Generate a qualitative model, if the two terms of the qualform of the hypothesis are observable terms, by copying the qualform into a new model frame stored under the active hypothesis,
and amend the tractability of the hypothesis appropriately, even if no model is generated.

**R5_SIMPLE_QUALFORM**
**Condition:**
There is an active qualitative hypothesis and all of its associated models have been tested,
and the terms of the hypothesis qualform are all measurable.
**Action:**
Generate a qualitative model by copying the qualform into a new model frame stored under the active hypothesis,
and amends the tractability of the hypothesis appropriately.

**R5_SIMPLE_HYPO_EQN**
**Condition:**
There is an active quantitative hypothesis and all of its associated models have been tested,
and there is an active paradigm that is not a combined experiment,
and all the equation's terms are measurable.
**Action:**
Generates a quantitative model by copying the equation into a new model frame stored under the active hypothesis,
and amends the tractability of the hypothesis appropriately.

**R5_MAIN_WORKER**
**Condition:**
There is an active quantitative hypothesis and all of its associated models have been tested,
and an active paradigm that is not a combined experiment,
and the equation does not have terms that are all measurable.
**Action:**
Attempt to generate a quantitative model from an hypothesis equation (or a term* frame) using background knowledge or definitions to infer alternative combinations of terms for unmeasurable terms, paying attention the necessary qualitative conditions, by substitution of the terms; several alternatives may possible, just 1 is considered and the rest are stored as term* frames in the partial_forms slot of the under the active hypothesis:
and amends the tractability of the hypothesis appropriately.

†The order of rules indicates their relative priority in conflict resolution.

STERN considers (R5_MAIN_WORKER). Generating quantitative models from such a quantitative hypothesis involves replacing terms that do not have corresponding measurable experimental parameters. STERN knows the T_V and T_DEN terms are not directly measurable, in this sense. This is just one of many examples of how experimental knowledge influences theoretical inferences. Two sources are investigated by STERN to replace them; (i) the definitions of the terms, and (ii) background knowledge.

Some terms are defined by combinations of other terms that are measurable. For example, T_V is related to T_D (distance) and T_TIME (time), by the equation:

$$(= \text{T\_V } (/ \text{ T\_D T\_TIME})) \; . \qquad \ldots (5.2)$$

STERN finds this equation (by looking for a filler in the *equation* slot the term's *T!* frame) and substitutes T_V, in Equation 5.1, for (/ T_D T_TIME). In general, this can only be done when certain necessary qualitative conditions obtain for the term to be eliminated (as specified in the *nec_qualform* slot of the T! frame). T_V can be replaced so long as T_V is constant. STERN knows (as one of its inputs) that the Aristotelian instantaneous acceleration is acceptable and thus the speed in the pendulum experiments is constant. (The Aristotelian law is later disconfirmed.) In this case STERN can perform the substitution.

The second way STERN can replace non-measurable terms is by appealing to background knowledge. STERN finds that the density term, T_DEN (of Equation 5.1), is one that maps onto a variable in the "spherical body" background knowledge relations; that is BODY_DEN. Background knowledge possessed by STERN is in the from of lists of *entry* frames (see Table 4.14). STERN chooses the entry that contains the BODY_DEN variable, and retrieves an equation that relates BODY_DEN to other variables from the entry frame. STERN converts this equation into theoretical terms and finds that T_DEN equals (/ T_W T_VOL) (the quotient of weight and volume). This formalism is substituted for T_DEN in equation 5.1.

The result of both substitutions is the model equation:

(= (/ T_D T_TIME) (/ T_W T_DEN). . . . (5.3)

STERN manufactures a model frame and this equation is placed in its *equation* slot.

However, the examination of background knowledge is really more complex than just described. STERN possesses more than one set of background knowledge and several entries may have variables that map on to the term to be substituted. Further, when the entry contains a qualform (rather than equation), the qualform must be converted into an equation before substitution can take place. Several equations may be generated from a single qualform. STERN only considers one possibility at a time, so all the remaining alternatives are stored for future reference (in *term** frames).

When STERN considers the other effective weight law:

(= T_V T_W*) , . . . (5.4)

the substitution of the T_V term is the same as before. However, the T_W* term (that stands for *effective weight*) is replaced using geometric background knowledge. This mimics Galileo's geometric-pictorial form of reasoning. Furthermore, STERN finds a qualform, rather than an equation, from the entry for T_W*. As qualforms cannot be used for in substitutions, STERN must find an equation based on the qualfrom. The program possesses several inference processes for converting qualforms into equations. For example, when the qualform has an *increase* predicate, STERN proposes a linear equation. The equation inferred is used for substitution, as before.

All new equations found by STERN are checked for new non-measurable terms that the substitution processes may have introduced. Such terms are themselves treated by recursively re-applying the same procedures to the new equation.

*5.3.1.3 Model Generation For Combined Experimental Paradigms*

The process we have just considered for generating models deals with more

typical experimental paradigms like pendulums. When STERN has an active *combined* experimental paradigm things are different (R5_COMB_COMPLEX_EQN). For example, STERN uses the combined inclined plane and projectile paradigms to confirm the law of free fall, given by:

$$(= T\_V\ (expt\ T\_H\ 2)), \qquad \ldots (5.5)$$

where T_H is height. STERN cannot eliminate T_V by substitution of its definition, as was the case for equation 5.1 above. The instantaneous acceleration law has, by now, been disconfirmed by STERN. So, the qualitative condition that the speed must be constant no longer obtains. This is why STERN uses the combined inclined plane and projectile experimental paradigm (see Chapter 8).

The are two *modes* in which STERN uses a combined experimental paradigm (see Chapter 3). In the *initial* mode, the manipulated input parameter is one belonging to the inclined plane, and the measured output parameter is from the projectile. In the *terminal* mode both the manipulated input and measured output are projectile parameters; with the inclined plane merely serving as a feeder for projectile. Variations in the model generation procedure are required for each of the two modes.

First, the initial mode: STERN initially applies Equation 5. to the inclined plane part of the combined experiment. This can be used to find the speed at the end of the plane for a given vertical height though which the ball drops. STERN then attempts to find a relationship describing the projectile part of the experiment that includes the T_V term. The program finds that:

$$(= T\_V\ (/\ T\_L\ T\_TIME)), \qquad \ldots (5.6)$$

where T_L is the projectile's horizontal distance and T_TIME is time. This asserts that the horizontal speed of the projectile is uniform. Hence, substituting out V from both equations 5.5 and 5.6 STERN obtains the model equation:

$$(= (expt\ T\_H\ 2)\ (/\ T\_L\ T\_TIME)). \qquad \ldots (5.7)$$

This equation is then treated like any other model equation by STERN.

Second, for terminal mode, STERN's generation of the model equation is almost the same as that just considered. The difference being that the free fall, Equation 5.5, is applied to the projectile rather than the inclined plane. By doing this STERN describes the shape of the projectile's path, just like Galileo.

*5.3.1.4 Qualitative Models Generation*

Generating qualitative models from hypotheses is a much simpler process for STERN than generating quantitative models.

To generate qualitative models STERN checks that all the terms in the hypothesis qualform have corresponding experimental parameters that are directly measurable in the active experimental paradigm (R5_SIMPLE_QUALFORM). When this is the case, STERN constructs a model frame and puts a copy of the qualform in its appropriate slot. However, when there are terms that are not measurable STERN checks that they are at least observable (see §4.2.4.1) (R5_COMPLEX_QUALFORM). The instantaneous acceleration law is one such qualform as it contains a term for speed. When the qualform has terms that are neither measurable nor observable STERN simple amends the hypothesis's measure of tractability to indicate that no model could be found.

Once STERN has found quantitative or qualitative models for the active hypothesis and experimental paradigm it uses each model to generate instances (*Generate instances* rules, Table 5.5).

### 5.3.2 Instance Generation

To generate instances STERN first chooses an experimental setup (R2_CHOOSE_SETUP). When considering the first Aristotelian, Equation 5.1, STERN selects one of the pendulum setups. STERN only uses two processes to generate instances; one for quantitative and one for qualitative models.

*5.3.2.1 Qualitative Instances*

STERN generates qualitative instances from qualitative models by first reproducing the model's qualform in an instance frame (R8_GEN_QUAL_INS-

## Table 5.5 GENERATE_INSTANCES Rules (RULES_8)

**R8_GEN_QUAL_INSTANCES†**
**Condition:**
There is an active qualitative model but no associated instances.
**Action:**
Construct a new instance, select terms to be the independent and dependent terms based on the manipulation and observation ease of the corresponding experimental experimental parameters (ie. matching kinds), and copy the model's qualform into the instance.

**R8_GEN_QUANT_INSTANCES**
**Condition:**
There is an active quantitative model but no associated instances.
**Action:**
A five stage process: (i) make all possible combinations of pair of terms found in the equation; (ii) eliminates all pairs that have one or more terms without experimental equivalents in the active experimental setup; (iii) removes all pairs that are trivially related by comparison with relevant background knowledge (iv) makes new instances for all the pairs of terms where the independent and dependent terms are chosen by reference to the manipulation ease of corresponding experimental parameters, and the values of the independent and fixed terms are also specified by interrogating experimental parameters; and (v) calculates the values of the dependent term by rearrange the equation so the term is on the left hand side, and substituting in the values.

†The order of rules indicates their relative priority in conflict resolution..

TANCES). STERN then specifies which terms are to be independent and dependent by examining the experimental parameters of the active experimental setup. The term that corresponds to the parameter that is most easily manipulated is chosen by the program to be the independent term. The other term from the qualform is then, by default, the dependent term.

*5.3.2.2 Quantitative Instances*

The generation of quantitative instances in STERN also employs the same correspondence procedure to choose its independent and dependent terms, but only after a number of other stages (R8_GEN_QUANT_INSTANCES). STERN's aim when generating instances is to find values for independent and dependent terms. The values of the dependent terms are calculated from an equation like 5.3 using independent term values, with the values of all other terms held constant. This is a three step process.

In the first step of the process, STERN finds all the possible combinations of pairs of terms from Equation 5.3:

(T_D T_TIME), (T_D T_W), (T_D T_VOL),

(T_W T_TIME), (T_VOL T_TIME), (T_VOL T_W). . . . (5.8)

Second, since an experimental setup need not instantiate all the parameters available in its paradigm, STERN removes any pairs from the list that include terms that are not measurable in the active setup. Third, further pairs are also eliminated if they are found to be trivially related. STERN does this by applying background knowledge to the experimental setup. Fourth, new instances are made employing the correspondence procedure just mentioned above; for the list of pairs (5.8) the (independent dependent) order of the terms is same as they are printed. Fifth, STERN calculates a number of evenly spaced independent term values within the range given by the maximum and minimum values of the term's corresponding experimental parameter. The values of the fixed terms are specified as the mid range

values of their corresponding parameter ranges. Sixth, magnitudes of the dependent term are calculated. STERN rearranges the Equation 5.3 so that the dependent term stands alone. Then the independent values and fixed term values are substituted into the equation to find the dependent values. That completes the steps in quantitative instance generation. All the generated instances are stored under the active model.

STERN has thus generated instances from the effective weight law (hypothesis) via models. This, in turn, completes the first stage of the confirmation processes. The next stage is the comparison of experimental test results and the instances.

## 5.4 COMPARISON WITH EXPERIMENTAL RESULTS

Here we will consider the third stage of the confirmation processes. During this stage STERN compares predictive instances and experimental test results in such a way that both predictive accuracy and noise in the experimental data are taken into account. But first, before comparisons can be made, STERN must obtain the experimental results.

### 5.4.1 The Design & Performance Of Experiments

STERN does two things when trying to obtain experimental results: (i) it designs an experimental test; and (ii) it performs the test (*Experiment* rules, Table 5.6).

STERN uses the active instance to design an experimental test from the given experimental setup. This ensures that the experiment will be relevant to the instance. Designing an experiment requires two things: (i) specifying the input-m, output and fixed parameters; and (ii) setting all their values, except the output. STERN does both by referring to the active instance (E_PREPARE_WITH_INSTANCE). The instance has terms and values that are equivalent to those required by the experimental test. For example, STERN uses the independent theoretical term to identify the input-m experimental parameter and gives the input-m identical values to those of the independent term.

## Table 5.6 EXPERIMENTER Rules (EXPT_RULES)

**E_COMBINED_EXPTS†**
**Condition:**
There is an active experimental setup and test,
and the experiment is the incplane+projectile or double pendulum combined experiment,
and the input-m and output parameters are of specific kinds.
**Action:**
Perform the combined experiment using the experimental test input values to calculate the output values with added noise.

The other five experimental performance rules have a similar form to the E_COMBINED_EXPTS rule except that different experimental setups and parameters are specified. The rules are:

**E_DOWN_INCPLANE_SPECIAL**
**E_PENDULUM_SPECIAL**
**E_SWING_PENDULUM**
**E_DOWN_INCPLANE**
**E_DOWN_PENDULUM**

**E_PREPARE_WITH_INSTANCE**
**Condition:**
There is an active instance but no experimental test,
and an active experimental paradigm.
**Action:**
For a qualitative instance the experimental setup is made with input and output parameters that have the same kinds and the instance independent and dependent terms.
For a quantitative instance the input, output and fixed parameters of the experimental test have the same kinds as the instance's independent, dependent and fixed terms, and the input and fixed parameter values are set according to their related instance term values.

†The order of rules indicates their relative priority in conflict resolution.

The performance of the experimental test is a job for STERN's experiment simulator (*Experimenter* rules, see Table 5.6). The simulator determines output values for all the input-m values. The output values are given an amount of noise that is realistic for experiments like the pendulum and inclined plane.

### 5.4.2 Assessing Predictive Accuracy

To assess how well his predictions matched experimental results Galileo employed various methods. For qualitative predictions, such as the one from the instantaneous acceleration law, it was sufficient for Galileo just to watch the motion of bodies when performing experiments. For his qualitative predictions, he compared the numerical values of terms directly and plotted graph like diagrams (eg. Drake, 1975; and Drake & MacLachlan, 1975).

STERN also uses different methods to assess qualitative and quantitative predictions (*Compare* rules, Table 5.7). However, STERN's methods are quite different from Galileo's, because STERN is not able to "see" the motion of bodies in experiments and does not have the ability to reason using diagrams.

Qualitative comparisons in STERN involves modelling the simple observations made during the performance of experiments. This is a three stage process (R12_TEST_QUAL_INSTANCE). First, since the results produced by the experiment simulator are quantitative, they have to be changed into qualitative forms before STERN can compare them with its instances. STERN infers qualforms from the input-m and output values of the test using regularity spotters (see Chapter 6). Second, the experimental qualforms are interpreted; that is, theoretical terms are substituted for their corresponding experimental parameters. Finally, STERN simply sees whether the instance qualform is amongst those just found. If it is, the instance is acceptable, if not the prediction has failed. The acceptability of the instance is set to zero or unity accordingly.

The comparison of quantitative instances and experimental tests by STERN is somewhat more involved. Basically, STERN determines how accurately the values

## Table 5.7 COMPARE Rules (RULES_12)

**R12_TEST_QUANT_INSTANCE**

**Condition:**

There is an active quantitative instance that has not been compared with an experimental test, and an active experimental test.

**Action:**

A two stage process: (i) check that neither the instance dependent term nor the experimental test output parameter are invariant; (ii) calculate the degree of match between the instance and experimental test according to equation (5.8) and set the instance acceptability.

**R12_TEST_QUAL_INSTANCE**†

**Condition:**

There is an active qualitative instance that has not been compared with an experimental test, and an active experimental test.

**Action:**

A three stage process: (i) infer qualforms from the input and output experimental test values; (ii) re-express these qualforms in theoretical terms by matching kinds and qualkinds; (iii) if one matches the instance qualform set instance acceptability to unity, otherwise zero.

†The order of rules indicates their relative priority in conflict resolution.

of the instance independent and dependent terms match up to the values of the input-m and output experimental parameters. There is no simple way to distinguish whether an instance and experimental test differ because the instance is a poor prediction or because the test data is noisy. Thus, STERN employs a single method to determine predictive accuracy and to take experimental noise into account. The technique comprises two main tests. (Bear in mind that the independent term and input-m parameter have identical sets of values.) The first test is the established *product-moment correlation technique*. It is applied to the dependent theoretical term and the output experimental parameter to find the degree to which two variables are related. A correlation coefficient, $r$, is found (where $-1 \leq r \leq 1$). The second test relies on the fact that the *ratios* of the dependent term and experimental output parameter would all be identical if there is a perfect match. The second tests measures how far actual instances are from this ideal, by seeing how the ratios vary with respect to the independent term. Ideally the ratio's values should fall on a straight line with zero gradient. Thus STERN employs *least squares analysis* to find the gradient, $m$, of the line that is the closest fit to all the ratios. When $m$ is zero the match is ideal, but when $m$ it is significantly far from zero the match is poor. For typical domain data, this second test is more than an order of magnitude more sensitive than product-moment correlation technique. However, the second technique may also find very poorly correlated data has $m \approx 1$ if it is fairly evenly scattered. Hence, STERN combines both techniques to overcome each other's weaknesses (R12_TEST_QUANT_INSTANCE). STERN calculates the value of predictive accuracy using the formulas:

$$\text{degree} = [(1 - |m|).r]^n , \qquad \ldots (5.9a),$$

when $r < R_{limit}$ ,

and $|m| > M_{limit}$ ;

otherwise $\text{degree} = 0$ . $\qquad \ldots (5.9b)$

Zero indicates no match between instance and experiment test, and a value near unity means good predictive accuracy. The user specified emphasis index, $n$, is used to further distinguish the degree by skewing values much less than unity towards zero (typically n=5). $R_{limit}$ and $M_{limit}$ are user specified parameters (typically both=0.5). When the absolute magnitude of $r$ is too low or $m$ too far from zero, the degree of match is set to zero. Thus the lowest degree calculated by Equation 5.9a will be no greater than 3 per cent (with n=5). STERN stores the value of predictive accuracy in the instance frame.

In our example involving the Aristotelian effective weight law, most of the comparisons between instances and experimental tests do not yield values that satisfy the two limits. The acceptabilities of the instances are consequently set to zero. However, for one instance, with T_D and T_TIME as the independent and dependent terms, the prediction of the motion on the inclined plane is rather good and the degree of match found by STERN was 0.966. We will see how these value are now used to help assess the acceptability of models and hypotheses, as we move on to the last stage of the confirmation process.

## 5.5 ASSESSING THEORETICAL KNOWLEDGE

We have seen how STERN compares instances and experimental results using a function that calculates values of predictive accuracy of instances. In this section we will consider the third, and final, stage of the confirmation process. During this stage STERN (i) assesses the acceptability of models with respect to the predictive accuracy of the instances generated from it and (ii) assesses the acceptability of hypotheses with respect to the acceptability of models.

In both cases STERN applies the acceptability criterion defined by Equation 4.1 (see §4.2.2.5) (R2_ASSESS_MODEL & R1_HYPO_ASSESS_WRT_MODELS). For example, when STERN was (dis)confirming the effective weight law, Equation 5.1, using the inclined plane paradigm, one model was generated and three

instances from that model. The instances were tested and just one was found to have a non -zero acceptability of 0.996. Thus the overall acceptability of the model is 0.332 (= {0.0 + 0.0 + 0.996} / 3). In addition to the inclined plane's model, one model was inferred for the pendulum paradigm, but was found to have an acceptability of zero. Thus the overall acceptability of the hypothesis is 0.166 (= {0.0 + 0.332} / 2).

Typically, once a hypothesis has been considered with all the existing experimental paradigms STERN simply moves on to the next hypothesis. However, if the hypothesis is a member of a group of related hypotheses, such as the law of free fall (IV, Table 4.1), STERN may perform an additional inference. Such groups of hypotheses are formed by STERN's strategy for generating new hypotheses from old (Chapter 7). The group of hypotheses has mutually exclusive members; that is, only one can be acceptable. Thus when STERN confirms the law of free fall all the others in the same group must be unacceptable (IV, Table 4.1). STERN makes the other hypotheses unacceptable by setting their acceptability measure to zero. This prevents STERN from unnecessarily attempting to confirm hypotheses it knows to be unacceptable by other means.

We have considered the three stages of STERN's confirmation strategy. The sheer number of knowledge types used and processes modelled gives an initial impression of the completeness of STERN. In the next section we will consider this in more detail and we will see how STERN has managed to overcome limitations of existing systems.

## 5.6 STERN ASSESSMENT ON CONFIRMATION

The quality of STERN's ability to confirm hypotheses can be assessed in two ways; (i) how well it models this aspect of the Galilean episode and (ii) how STERN compares with previous discovery systems that perform similar types of processes.

### 5.6.1 Completeness Of The Confirmation Strategy

In Chapter 4, we saw that STERN models the whole of the Galilean episode rather well. The confirmation strategy is called upon by STERN four times: first to disconfirm the Aristotelian laws; second to test the hypotheses that were generalized from experimental results; third to examine new hypotheses that have been generated from old hypotheses; and fourth after new experimental paradigms had been invented. In each case STERN successfully judged as acceptable the hypotheses that Galileo thought were true. Further, no hypotheses were considered as acceptable when they were not generally valid - STERN did not produce false positive hypotheses.

The overall success of the confirmation strategy argues for the acceptability of the criterion of acceptability assumed in STERN. The criterion is basically one of explanatory breadth, but it does not just consider numbers of items of evidence. The acceptability of hypotheses is assessed in terms of the relative successes of its models, that are in turn assessed in terms of the relative success of its instances. The instances are compared directly with experimental tests. In this way the law of free fall was found to be the only generally-acceptable hypothesis in the domain.

Chapter 3 highlighted the importance of both qualitative and quantitative reasoning in Galileo's inferences. The modelling of the confirmation strategy acknowledges this by dealing with both quantitative and qualitative hypotheses, models and instances. Although STERN considers individual hypotheses in turn, previously-assessed qualitative hypotheses may influence the way later quantitative hypotheses are assessed. For example, the instantaneous acceleration law was initially thought to be valid and thus permitted speed terms to be substituted for distance and time in equations early on. However, later in the episode the law was shown to be unacceptable and other means were required to deal with the speed term.

The results provided by the experiment simulator contain a realistic amount of noise. STERN is able to cope with this adverse influence on experiments during the comparison of instances and experimental tests. The function that determines the predictive accuracy of an instance takes noise into account. The greater the noise the less the degree of acceptability. Dealing with noise in experimental data is one of the aspects of determining the reliability of the experiments.

The importance of the communication between theory and experiments during the confirmation strategy is clearly seen in STERN. When choosing which hypothesis to test it is necessary to know if the hypothesis has already been considered with all the manufactured experimental paradigms. The generation of models and the making of predictions relies on the knowledge of the experimental paradigm and experimental setup that is to be accounted for. The design and performance of experimental tests needs information about the predictive instance so that the two can eventually be compared.

STERN models the ability of scientists to recognize mutually exclusive or contradictory theories and to differentially assess their acceptabilities when just one has been successfully tested. A simple mechanism groups together hypotheses that are obtained simultaneously from a single inference. STERN uses information during the confirmation strategy to reduce the size of the hypothesis space. For example, when the law of free fall was found all the other hypotheses with similar terms were known to be unacceptable. Their measures of acceptability were set to reflect this and from then on they were simply ignored.

In general, STERN is well able to model the confirmation of hypotheses. As we will see its abilities go beyond those of previous systems.

### 5.6.2 Advances On Previous Work

In Chapter 2 we saw that most previous computer models of scientific discovery have tended to concentrate on the generalization of instances into models (eg. all the programs of the BACON school), so only a few programs can be compared directly

with the confirmation strategy in STERN. However, none of these follow a discovery path anything like STERN's and none model a domain like the Galilean one. MetaDendral, SDDS and KEKEDA will be considered in detail.

MetaDendral (Buchanan & Feigenbaum, 1978) does possess a representation that can be considered as a hypothesis from which models are generated. However, this hypothesis is simply the most general rule that describes how a molecule may fragment. Models in this view, are the child rules that are successive specializations of a original parent rule, as generated by the RULEGEN subprogram. Thus the hypothesis merely serves as the root of the search tree of possible fragmentation rules, without being related to its children and grand children in any substantive sense. In STERN, however, many different hypotheses (not just one) are considered. Furthermore, the structure of model equations depends on the particular form and terms of hypotheses, and instances are specific instantiations of a model.

The SDDS model of discovery (Klahr & Dunbar, 1988) has not been implemented in a running program, however it does propose processes that are like those instantiated in STERN, including a subset of those employed in the confirmation strategy. One possible pattern of discovery permitted by SDDS's processes hierarchy starts by fully specifying a proposition using *prior knowledge* or *old outcomes* (see Figure 2.2). The *testing* of the proposition involves: (i) designing an experiment by *focusing* on some aspect of it and *choosing* and *setting* variables; (ii) making a *prediction*; (iii) *running* the experiment and *observing* the outcome; and (iv) *matching* the prediction and outcome. Finally, the *decision* is made whether to accept or reject the proposition by *reviewing outcomes*. This pattern maps neatly onto STERN's processes of; generating instances from a model, designing and performing an experimental test, and comparing the instance and test result to assess the model (see Table 2.2). However, Klahr and Dunbar (1988) do not have quantitative measures of proposition acceptability.

A similar situation occurs between STERN and KEKEDA (Kulkarni & Simon, 1988) as just described for STERN and SDDS. KEKEDA simulates the discovery of the Urea Cycle by formulating a model that involves several different reactions in sequence. Each of these reactions, or propositions, is formulated and tested in turn by making predictions about the nature of the reaction with specific participating substances. The confidence that the reaction is correct is determined in part by the number of correct predictions made from it. All this is similar to that part of STERN's confirmation process that just focuses on model confirmation. To map KEKEDA into STERN read; model for reaction, prediction for instance, substance for term or parameter, and confidence for measure of acceptability. The essential point is that KEKEDA and SDDS are subsumed by STERN.

A limitation of STERN's confirmation strategy is that it only considers one hypothesis at a time. Sleeman *et.al.*'s (1989) Architecture for Theory-Driven Scientific Discovery notes that well established core theories in conjunction with less general weak theories may both play a part in the generation of new theories. The work on problem solving in the domain of physics problems (eg. Larkin *et.al.*, 1980; Bundy *et.al.*, 1979; and Luger, 1980) shows that scientists can use several principles (hypotheses) in combination to account for a particular situation. When considering more than one hypothesis at a time, heuristics like those found in AM and EURISKO (Davis & Lenat, 1982; Lenat 1983; Lenat & Brown, 1984) that consider the "interestingness" and "worth" of concepts may be required to choose between competing hypotheses with the same degree of acceptability. The ability to deal with multiple hypotheses at one time may be needed by STERN in the future, if it is to cope with more complex theories. However, in the Galilean domain dealing with one hypothesis at a time approach is sufficient.

### 5.6.3 Conclusion

Clearly STERN's confirmation strategy is compatible with previous successful scientific discovery systems and proposed models. However, STERN also overcomes some of their limitations. In particular:

- All the types and levels of scientific knowledge posited by the framework are represented.
- Processes operating on all the types and levels of knowledge are modelled including many that require communication between theory and experiment.
- The acceptability of hypotheses and models is assessed in terms of breadth of experimental evidence as indexed by the acceptability and numbers of models and instances, respectively.
- Noise in experimental results is dealt with whilst predictive accuracy is assessed using a function that considers all the available data at once.
- Many different types of communication occur between all levels of the theory and experiment components in a research programme.
- Background knowledge was used to help in the generation from hypotheses to models, and from model to instances.
- Mutually exclusive hypotheses are considered as groups to improve the efficiency of the discovery processes by dramatically cutting down the searched space.

Many of these abilities are are not unique to STERN's confirmation strategy but range across all of the program's main processes, as we will see in the following chapters.

In this chapter we have seen how STERN assesses the acceptability of existing hypotheses by generating models and making predictive instances. But where do the hypotheses come from in the first place? The next two chapters will answer this question. Specifically, in the following chapter we will see how STERN employs a strategy of experimental-led inductive generalization to find hypotheses.

# Chapter 6

# Generalization From Experiments To Hypotheses

## 6.1 INTRODUCTION

Galileo's original intention was to confirm the Aristotelian laws by comparing predictions made from them against experimental results. However, as we saw in Chapter 5, the laws were found to be unacceptable and he was left with a small number of experimental results. In the next stage of this episode of discovery Galileo found new hypotheses by generalizing experimental results. This involved designing and performing a wider range of experiments than before to establish a body of experimental results. These results were then generalized to form hypotheses. Such a generalization strategy is an important constituent of the overall discovery process.

The generalization of empirical data into higher level theoretical knowledge is the area in which most previous work on scientific discovery has focussed (see e.g. Langley *et.al.*, 1987; Thagard, 1988a; Falkenhainer & Michalski, 1986; Gerwin, 1973; Qin & Simon, 1990). In terms of the framework for scientific discovery all such work has typically concentrated on the formation of a model from a set of instances.

STERN instantiates the inductive process in a more complete manner. It deals with the selection of experimental paradigms, the design and performance of experimental tests and the interpretation of the results into instances. The instances are generalized to form models and the models generalized to form hypotheses. Whereas previous systems have only found models, STERN finds a large variety of quantitative and qualitative hypotheses. These hypotheses give STERN a deep

understanding of the domain; for example, the qualitative hypotheses tell STERN which terms are relevant or irrelevant to the characterization of the phenomena. During the generalization stage of of its modelling, STERN also makes a "genuine" discovery - the law governing the period of swing of pendulums.

STERN chooses to invoke the strategy that generalizes experimental results into hypotheses when all existing hypotheses have been tested and shown to be unacceptable but experimental paradigms remain to be accounted for (rule R0_START_INDUCE of *Strategy chooser*).

This chapter considers the experimental-led inductive generalization to hypotheses, beginning with an overview of the process (§6.2); this is followed by a detailed examination of the stages in the strategy (§6.3 to §6.5). Finally, comparisons are made between this aspect of STERN's abilities and previous discovery systems (§6.6).

## 6.2 STAGES IN MODELLING GENERALIZATION

The strategy employed by STERN to generalize from experimental results into hypotheses can be summarized diagrammatically (see Figure 6.1). The boxes in Figure 6.1 are levels of knowledge as given by the framework and the circles are inference processes. STERN uses the *models into hypotheses* class of rules to control the process (Table 6.1).

The process starts with the selection of an experimental paradigm, such as the pendulum. It has not been accounted for by any acceptable hypotheses stored on the theoretical side of the research programme (R3_CHOOSE_PARADIGM). From the paradigm, an experimental setup is chosen; such as the setup allowing the pendulum to swing freely in a periodic manner. This in turn permits STERN to design various experimental tests with different parameters as the INPUT-M and the OUTPUT. For example, one test may involve manipulating the length of the pendulum to see how

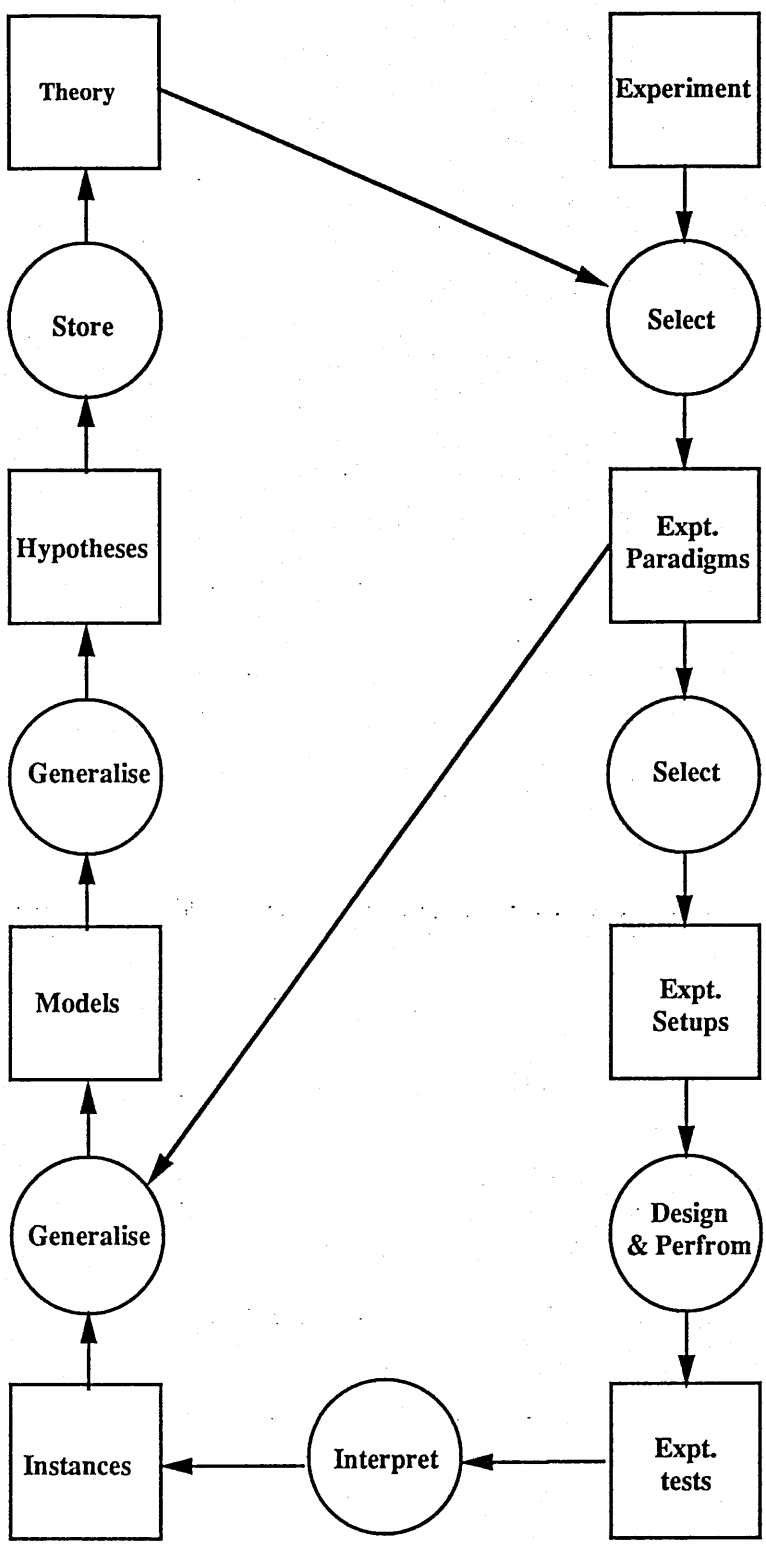



Figure 6.1 STERN's Generalisation Process

## Table 6.1 MODELS INTO HYPOTHESES Rules (RULES_3)

**R3_CHOOSE_PARADIGM***

**Condition:**

There is no active experimental paradigm,
and the active hypothesis (if there is one) is frame,
and there are stored experimental paradigms for the research programme of which at least one has not been acceptably accounted for by previous hypotheses or is being considered with active hypotheses now.

**Action:**

Construct and make active an hypothesis frame for the storage of models if none already exists,
and make active the manufactured experimental paradigm that has the greatest product of number of setups and manufacture ease that has not been acceptably accounted for by previous hypotheses,
and add the name of the paradigm to the list of names stored under the active hypothesis

**R3_GET_MODELS**

**Condition:**

There is an active experimental paradigm,
and no active model(s).

**Action:**

Call the PS with *Instances into models* (RULES_7) to obtain models from experimental setups and instances to generalize the models into hypotheses.

**R3_STORE_MODELS**

**Condition:**

There an active list of models.

**Action:**

Store the models under the active hypothesis (ie. copy the list of models and then clear them).

**R3_GENERALISE_MODELS**

**Condition:**

There is a single active hypothesis with associated models.

**Action:**

Call the PS with the domain specific rules in generalise_models slot of the theory frame (i.e., *Generalize models*, RULES_10) to generalise the models into hypotheses.
On return make inactive any current models or experimental paradigms

*The order of rules indicates their relative priority in conflict resolution.

the period of swing varies. STERN then performs the experiment to obtain experimental results (using its experiment simulator). To be able to work with the results theoretically STERN must convert them into instances. This interpretation process mainly involves finding the corresponding theoretical terms for particular experimental parameters. Many instances are obtained from many experimental tests. Once all the experimental tests for a given experimental paradigm have been performed and interpreted into instances, STERN can go about generalizing the instances to form models. Both qualitative and quantitative models are inferred and the experimental paradigm to which they applied is noted. One of the models that STERN finds at this stage is the law relating the length of a pendulum to its period of swing. In the same way STERN obtains many other models for the same experimental paradigm and stores them (R3_GET_MODELS & R3_STORE_MODELS). The whole processes is repeated for the other available experimental paradigms. STERN finally generalizes the many models associated with specific experimental paradigms into hypotheses (R3_GENERALISE_MODELS).

STERN designs many experimental tests for each experimental setup. Thus it employs repeated cycles of the processes that perform the testing and interpretation of experimental results into instances (i.e. the *tests into instances* rules, Table 6.2). Similarly, an experimental paradigm may have more than one experimental setup. Hence, cycles of processes are also associated with each setup (i.e. the *instances into models* rules, Table 6.3). One of theses cycles consist of four processes in sequence: (i) selecting an experimental setup; (ii) designing experimental tests; (iii) performing all the tests and interpreting their results into instances (i.e. sets of previous cycle just described); and (iv) generalizing the instances into a model. Hence, in overview, STERN's generalization strategy consists of cycles of processes which employ sub-processes that also comprise cycles of nested processes at a deeper level (see also, Figure 4.2).

This rather complex corpus of processes can fortunately be broken down into

## Table 6.2 TESTS INTO INSTANCES Rules (RULES_11)

**R11_INTERP_TO_INST***
**Condition:**
There is no active instance,
and an active experimental test that has been performed.
**Action:**
Call the PS with the rules in the theory frame interp_expttest slot (i.e., *Interpret*, RULES_6) to interpret the experimental test into an instance.

**R11_PERFORM_EXPT_TEST**
**Condition:**
There is an active experimental setup,
and no active instance,
and an active experimental test yet to be performed (ie. the output parameter values are yet to be found).
**Action:**
Call the PS with the rules in the experiment frame perform_expttest slot (i.e., *Experimenter*, EXPT_RULES) to carry out the experiment.

*The order of rules indicates their relative priority in conflict resolution.

stages. We can think of the overall generalization as having three stages: (i) obtaining experimental test results from experimental paradigms; (ii) interpreting test results into instances; and (iii) generalizing instances into hypotheses via models. We will consider details of the generalization in STERN in terms of the three stages in the next three sections. STERN uses seven classes of rules to model this strategy (Tables 6.1 to 6.5 & 6.7). The priority of the rules is given by their order (in each table).

## 6.3 OBTAINING EXPERIMENTAL RESULTS

As we saw in the summary of the whole generalization strategy (in the previous section), just obtaining experimental results requires STERN to: (i) choose an experimental paradigm and setups; (ii) design experimental tests; and (iii) perform the tests to obtain the results.

### 6.3.1 Selecting Experimental Paradigms & Setups

STERN chooses an experimental paradigm from amongst those that are *available* (see Chapter 8). The exact choice depends on (i) whether a paradigm has been adequately accounted for by an existing hypothesis and (ii) a pragmatic value that is calculated for each paradigm. For a given paradigm this pragmatic value is the product of the number of setups and their ease of manufacture. The experimental paradigm with the lowest value is chosen (R3_CHOOSE_PARADIGM). This just happens to be the pendulum paradigm, because it is so easy to manufacture its setups. At this point an active hypothesis is also constructed as a repository for models found later on.

For the active experimental paradigm STERN now needs to choose an experimental setup, with the long term aim of obtaining models (R3_GET_MODELS invokes the *instances into models* rules, Table 6.3). When selecting an experimental setup STERN simply chooses the first that has not already been considered, provided that it does not need to be part of a combined experiment

## Table 6.3 INSTANCES INTO MODELS Rules (RULES_7)

**R7_OBTAIN_EXPTTESTS***

**Condition:**

There is an an active experimental paradigm,
and an active experimental setup with no associated (stored) experimental tests,
and no active experimental tests,
and no active instance.

**Action:**

Make a list of active experimental tests by: (i) making a list of experimental parameters preferring those from the active setup over the active paradigm; (ii) making all combinations of pairs of parameters without duplication (iii) eliminating pairs that are trivially related by examining pertinent background knowledge; (iv) making an experimental test for each pair in turn, choosing the input-m and output parameters according to their values of ease of manufacture, and setting the range of input-m and fixed values using their maximum and minimum permitted values given by the fillers of those slots in the parameter.

**R7_MAKE_INSTANCES**

**Condition:**

There is an an active experimental paradigm,
and an active experimental setup,
and active experimental test that is a frame.

**Action:**

Call the PS with *Tests into instances* (RULES_11) to obtain instances from experimental tests.

**R7_EXPTTEST_PREFERENCES**

**Condition:**

There is an an active experimental paradigm and experimental test,
and the experimental setup does not have any associated tests.

**Action:**

Store all experimental tests that have speed or time as input-m or output parameters (ie. are actually concerned with accelerated motion) in the setup's list of tests.

**R7_STORE_INSTANCES**

**Condition:**

There is an active experimental paradigm
and active instance(s).

**Action:**

Add the instance(s) to the list of instances in the models instance slot,
and deactivate the instance(s).

**R7_CHOOSE_EXPTTEST**

**Condition:**

There is an an active experimental paradigm,
and an active experimental test with associated experimental tests,
and no active experimental tests.

**Action:**

Remove the and make active the first experimental test associated with the experimental setup.

continued . . .

## Table 6.3 Continued INSTANCES INTO MODELS Rules (RULES_7)

**R7_CHOOSE_SETUP***

**Condition:**

There is an an active experimental paradigm,
and no active experimental setup,
and an active model that is a frame,
and experimental setups that have not yet been tried that do not have to be part of a combined experiment.

**Action:**

Construct an active model if one does not already exist,
and make active an experimental setup that has not already been covered by the model that does not have to be part of a combined experiment,
and add the setup's name to the model's list of such names.

**R7_GENERALISE_INSTANCES**

**Condition:**

There is an an active experimental paradigm,
and an active model that is a frame with associated instances,
and their is no active instance.

**Action:**

Call the PS with the rules stored in generalise_instance slot of the theory frame (i.e., *Generalize instances*, RULES_4) to generalise instances into models.

*The order of rules indicates their relative priority in conflict resolution.

(R7_CHOOSE_SETUP). At this point an active model is constructed as a repository for instances that are made later on. Now that an active setup exists STERN uses it to design experimental tests.

### 6.3.2 Designing Experimental Tests

The procedure STERN employs to design experimental tests from an active setup involves five stages. First, a list of all the experimental parameters is made, preferring those from the active setup over the active paradigm (R7_OBTAIN_EXPTTESTS). For the pendulum setup these paradigms are ones for *time*, *distance*, *height*, *length*, *weight*, *volume*, *size*, *angle* and *speed*. Second, all the combinations of pairs of parameters are made without duplication; 36 for the pendulum setup. Third, pairs of parameters are eliminated if they are known to be trivially related using pertinent background knowledge. For the pendulum setup, those pairs with any combination of *distance*, *height*, *length*, *size*, and *angle* are eliminated using the geometric knowledge. The (*weight volume*) pair is eliminated using the relations for spherical bodies, leaving 27 pairs. Fourth, experimental tests are made for each pair in turn (i.e. their frames constructed). The parameters to be the input-m and output are chosen from the pairs according to their relative ease of manipulation and observation (e.g. for the (*time size*), input-m = *size*, output = *time*). A series of evenly spaced values are calculated for the input-m parameter using its maximum and minimum permitted values. The magnitude of each fixed parameter is set to its mid-range value. (The series output parameters values are to be found by performing the test.). Fifth, those experimental tests that have no relevance whatsoever to the domain are weeded out (R7_EXPTTEST_PREFERENCES). When modelling the Galilean domain STERN removes tests that have nothing to do with motion; that is, those that do not include either a *time* or a *speed* parameter as an input-m or an output. Thus a total of 15 experimental setups are finally designed for the pendulum setup.

In this way STERN makes many experimental tests. Each experimental test is considered in turn, with the aim of obtaining instances (R7_MAKE_INSTANCES invokes the *Tests into instances* rules). Each experimental test is performed and its results interpreted into instances.

### 6.3.3 Performing An Experimental Test

STERN performs an experiment on active tests using its experiment simulator (see see §4.5) (R11_PERFORM_EXPT_TEST invokes the *Experimenter* rules). The details of this processes were considered when we looked at the confirmation strategy (see §5.3.2). The net effect of performing the test is that the output parameter values of the experimental test are found. STERN's test results may now be interpreted into instances.

## 6.4 INTERPRETING EXPERIMENTAL RESULTS

STERN attempts to find both quantitative and qualitative instances from the experimental test results (R11_INTERP_TO_INST invokes *Interpret*, Table 6.4).

The interpretation to instances with quantitative data has two parts: (i) finding corresponding theoretical independent and dependent terms for the input-m and output experimental parameters; and (ii) copying their respective series of magnitudes (R6_SIMPLE_TRANSFER). An instance is constructed for each set of quantitative data. The acceptability of the instance is set to 1 to reflect the fact that the magnitudes of the instance were obtained directly from experiment.

When STERN looks for qualitative instances it is in effect modelling the observations (rather than measurements) that Galileo made during the performance of experiments. Thus STERN must convert the quantitative experimental data into qualforms (R6_FIND_QUALFORMS). STERN recognizes many qualforms (Table 4.13) and possesses functions that spot whether such qualforms apply to two related series of values. For example, the test for the relevance of the INCREASE qualform requires that all magnitudes in both series are monotonically increasing.

## Table 6.4 INTERPRET Rules (RULES_6)

**R6_SIMPLE_TRANSFER***

**Condition:**

There is an list of active instances,
and an active experimental setup that has input-m and output parameters that are measurable and are not unrelated (ie. the independent and steady qualforms do not apply to the input-m and output).

**Action:**

Construct a new instance using the experimental test parameters to find the corresponding theoretical terms to fill the appropriate slots,
and set the degree of acceptability slot of the instance to unity,
and make inactive the experimental test.

**R6_FIND_QUALFORMS**

**Condition:**

There is an active experimental setup,
and no active instance.

**Action:**

Identify qualforms based on the input-m and output parameters of the experimental test converted into their corresponding theoretical terms,
and construct an instance for each qualform with the degree of acceptability set to unity.

*The order of rules indicates their relative priority in conflict resolution.

These functions are applied to the lists of values from the experimental test input-m and output parameters. If a function finds an applicable qualform STERN constructs an instance for the qualform.

The instances that STERN finds by interpreting each experimental test result are stored (R7_STORE_INSTANCES). When all the experimental tests for an active experimental paradigm have been considered (via its setups), STERN starts to generalise the stock of instances into higher theoretical knowledge beginning with models.

## 6.5 GENERALIZING INSTANCES INTO HYPOTHESES

STERN does the generalization of instances into hypotheses in two stages: (i) models are found from the instances; (ii) the models are generalized to form hypotheses.

### 6.5.1 Instances Into Models

When STERN has finished performing and interpreting experimental tests from a particular experimental paradigm there will be many stored instances (in the model acting as a repository). STERN generalizes the instances into models (R7_GENERALISE_INSTANCES invokes *Generalize instances*, Table 6.5). STERN finds qualitative models from qualitative instances, and quantitative models from instances with numerical data.

The aim of STERN when generalizing qualitative instances into models is to obtain model-qualforms that validly apply to each of the experimental setups, under the active experimental paradigm (R4_MODEL_QUAL). STERN groups together instances with identical qualforms. For example, all instances with (INCREASE T_V T_H) are put together. Those groups that have fewer than a user-specified number of qualforms are rejected. The qualform that is common to each group is used directly in the construction of a new model, with acceptability and tractability measures set according to the number of instances in the group.

## Table 6.5 GENERALIZE INSTANCES Rules (RULES_4)

**R4_MODEL_QUAL***
**Condition:**
There is an list of active instances some of which have qualforms.
**Action:**
Generalization of instances into qualforms by: (i) finding all instances that have qualforms; (ii) making a list of just qualforms from instances; (iii) constructing a model for those qualforms that occur X or more times, where X is a user specified integer.

**R4_MODEL_EQNS**
**Condition:**
There is an list of active instances,
and some of the instance have dependent terms.
**Action:**
For each quantitative instance make a model by: (i) obtaining the index of the power function that best fits the independent and dependent terms by finding the gradient of line of the graph of the logs of the terms using the least squares fit technique; (ii) finding a fraction that is equal to the index within a user specified accuracy, and maximum integer range of the numerator and denominator; (iii) when a rational index is found construct a model with an equation in terms of the independent and dependent using the fractional index, with the tractability and acceptability set accordingly,
and make the instance(s) inactive.

**R4_PREPARE**
**Condition:**
There is an active model that is a frame with associated instances,
and no active instance(s).
**Action:**
Remove the instances stored under the active model and make them active.

*The order of rules indicates their relative priority in conflict resolution.

However, when modelling, the Galilean episode STERN has few experimental setups for each experimental paradigm. Thus the user-defined number of instances per group is set to one: that is, all qualitative instances become models with qualforms. This simply reflects that fact that the experiments in this domain do not have the same degree of complexity found in many other fields. However, this does not mean that there is no difference between instances and models in STERN. Some models are generalized from more than one instance and quantitative instances are very different from quantitative models, as we will now see.

The generalization to quantitative models in STERN involves finding an equation relating together the instance independent and dependent terms, given their lists of values (R4_MODEL_EQNS). This is exactly the sort of task that BACON.1 (Langley *et.al.*, 1987) performs. However, a rather different approach is adopted in STERN.

Consider the power function of two terms. A graph of the terms with values plotted as logarithms will be a straight line whose gradient is equal to the power, or index, of the function. Thus given a set of data the best fitting power function can be found by plotting a log-log graph and finding the straight line that intersects most points. This procedure is exploited by STERN in its three stage process for finding equations. First, the index of the power function is found. Graphs cannot be plotted by STERN so the *least squares fit* method is applied to the logarithms of the values of the instance's independent and dependent terms. The result is an index, $n$, whose value is a positive real number. However, Galileo only considered simple power equations with rational indices. Similarly, in STERN the second stage of the processes involves finding a fraction of equal magnitude to the index. Let the fraction be $p/q$, where $q$ is the denominator and $p$ the numerator. Now, STERN only considers values of $p$ and $q$ that are integers below a certain user-specified limit (typically 3). So a search of the rational index takes place in the space of

fractions defined by the different combinations of $p$ and $q$. A *deviation* test determines whether $p/q$ is sufficiently close to $n$ by testing if the difference between $n$ and $p/q$ is less than some user-specified value (typically 0.015). If and when a suitable value of $p/q$ is found, the third stage constructs a model that has a power equation relating the two terms with $p/q$ as the index. When the index is itself an integer (e.g. $2p=q$) the standard form of equation is used with multiples of one term. For example, STERN finds the equation relating the period (T_TIME) of swing of a pendulum to the length of its chord (T_S):

(= T_S (* T_TIME T_TIME)). . . . (6.1)

A model is constructed for the equation, and the tractability and acceptability of the model are set. In the case when no rational index is found, no model is constructed.

Overall this procedure differs in three significant ways from the technique employed in BACON.1 (Langley *et.al.*, 1987). First, the first stage of STERN's procedure uses just the one process to find the relation between the terms, in effect condensing BACON's multiple applications of regularity spotters into a single operation. Second, the procedure is more efficient because the search processes in the second stage only examines the space defined by the integer values of $p$ and $q$, where a new state is generated by adding (or subtracting) 1 to $p$ or $q$. BACON's search, however, is in a space of terms; each state is generated by finding a new term but also calculating a whole new series of values for that term. Third, the least squares technique that initially finds the index also has the effect of averaging out any noise in the data. BACON on the other hand employs a technique that examines whether each individual value is within a band centred on the mean of a constant term.

STERN discovers many models with qualforms and equations (see Table 6.6). Thirty three models are found that apply to the pendulum paradigm, and 21 that apply to the inclined plane. The qualforms with REPEAT+ and REPEAT- predicates only occur under the pendulum paradigm, indicating that repetitive motion (i.e. the

## Table 6.6 Model Qualforms And Equations For Two Experimental Paradigms

### PENDULUM

```
(FROM_ZERO T_V T_@)
(REPEAT+ T_V T_S)
(STEADY T_V T_VOL)
(STEADY T_V T_W)
(REPEAT+ T_V T_L)
(REPEAT+ T_V T_H)
(REPEAT+ T_TIME T_@)
(REPEAT- T_TIME T_VOL)
(REPEAT+ T_TIME T_W)
(FROM_ZERO T_TIME T_L)
(FROM_ZERO T_TIME T_H)
(INCREASE T_V T_@)
(REPEAT- T_V T_VOL)
(INCREASE T_V T_L)
(INCREASE T_TIME T_@)
(INCREASE T_TIME T_L)

(REPEAT+ T_V T_@)
(REPEAT+ T_V T_VOL)
(REPEAT- T_V T_W)
(FROM_ZERO T_V T_L)
(FROM_ZERO T_V T_H)
(FROM_ZERO T_TIME T_@)
(REPEAT+ T_TIME T_S)
(STEADY T_TIME T_VOL)
(STEADY T_TIME T_W)
(REPEAT+ T_TIME T_L)
(REPEAT+ T_TIME T_H)
(INCREASE T_V T_S)
(INCREASE T_V T_W)
(INCREASE T_V T_H)
(INCREASE T_TIME T_S)
(INCREASE T_TIME T_H)

(= T_S (* T_TIME T_TIME))
```

### INCPLANE (inclined plane)

```
(STEADY T_V T_VOL)
(STEADY T_V T_W)
(INCREASE T_V T_L)
(INCREASE T_V T_H)
(INCREASE T_V T_D)
(STEADY T_TIME T_W)
(FROM_ZERO T_TIME T_L)
(FROM_ZERO T_TIME T_H)
(FROM_ZERO T_TIME T_D)

(= T_D (* T_TIME T_TIME))
(= T_L (* T_TIME T_TIME))

(INCREASE T_V T_VOL)
(FROM_ZERO T_V T_L)
(FROM_ZERO T_V T_H)
(FROM_ZERO T_V T_D)
(STEADY T_TIME T_VOL)
(INCREASE T_TIME T_W)
(INCREASE T_TIME T_L)
(INCREASE T_TIME T_H)
(INCREASE T_TIME T_D)

(= T_H (* T_TIME T_TIME))
```

swinging) is unique to that experiment. Equation 6.1, found under the pendulum paradigm, correctly describes the relationship between the period of swing of a pendulum and its length (size). This is arguably a "genuine" discovery made by STERN, because there was no intention that this model should be found on the part of the programmer. However, with hindsight it is not completely unexpected given STERN's realistic experiment simulator. Notice that the three model-equations found under the inclined plane paradigm have the same form. This is not surprising because distance, height and length are related by the geometry of the inclined plane. Of the three equations, the two covering distance & time, and height & time, are expressions of Galileo's law of free fall. However, in this form these equations are not completely general.

Thus many models are inferred by STERN for each experimental paradigm. The models are stored (R3_STORE_MODELS) and further experimental paradigms are considered. Once all available paradigms have been considered an attempt is made to generalize the models into hypotheses.

### 6.5.2 Models Into Hypotheses

The generalization of models to form hypotheses also considers qualitative and quantitative knowledge separately (R3_GENERALISE_MODELS invokes *Generalize models*, Table 6.7).

The generalization of qualitative models into hypotheses by STERN is a three stage process, whose aim is to obtain hypotheses that range over a sufficiently large number of experimental paradigms (R10_HYPO_QUALS). First, all qualitative models are grouped according to similar qualforms. Second, those groups that have one model for every available experimental paradigm are chosen; that is, the one type of qualform common to the group is applicable across all the experimental paradigms. In STERN there must be two models to the group as two experimental paradigms have been considered. In the third stage, hypotheses are constructed that contain a qualform from each chosen group. The acceptability and tractability of each

## Table 6.7 GENERALIZE MODELS Rules (RULES_10)

**R10_HYPO_QUALS***
**Condition:**
There is an active list of models some of which are qualitative.
**Action:**
Generalization of qualitative models with into qualitative hypotheses by: (i) grouping together all models with similar qualforms; (ii) choosing groups that have more than a user specified number of models that account for different experimental paradigms; and (iii) constructing a hypotheses for each of the groups using the group's qualform and set the acceptability and tractability appropriately.
And remove qualitative hypotheses from the current list of models.

**R10_HYPO_EQNS**
**Condition:**
There is a list of active quantitative models.
**Action:**
For each quantitative model a model is made with the same equation and the acceptability and tractability are set appropriately.

**R10_PREPARE**
**Condition:**
There is an hypothesis that is a frame with associated models,
and no active model.
**Action:**
Remove the models from the hypothesis and make them active.

*The order of rules indicates their relative priority in conflict resolution.

hypothesis is set according to the number of models in the group. In the modelling of the Galilean episode, STERN reduces the 50 qualitative models down to just 12 hypotheses (II, Table 4.1). This demonstrates the importance of modelling the hypothesis and experimental paradigm levels of scientific research programmes. Previous discovery systems have tended not to do so and have consequently had to deal with explosive numbers of models.

The twelve qualitative hypotheses that STERN has found contain some very valuable information (II, Table 4.1). For example, the qualform (STEADY T_TIME T_VOL) indicates that T_TIME is unrelated to T_VOL, because T_TIME is constant as T_VOL is varied. Now as the qualforms:

(STEADY T_TIME T_VOL) (STEADY T_TIME T_W)

(STEADY T_V T_VOL) (STEADY T_V T_W), . . . (6.2)

are all found in hypotheses, this means that the terms T_VOL and T_W (volume and weight) are not relevant to the characterization of naturally accelerated motion. By a similar argument using the INCREASE and FROM_ZERO qualforms, and STERN's other qualitative hypotheses (II, Table 4.1), we can see that STERN knows that T_V, T_TIME, T_H and T_L (speed, time, vertical and horizontal distances) are important to the characterization of the phenomena.

A simpler process performs the generalization of quantitative models into hypotheses. The desirability of quantitative hypotheses is contrasted by the comparative rarity of quantitative models, thus it seems worthwhile considering all models with equations as potential hypotheses. Hence, STERN simply constructs an hypothesis using the model's equation to fill the equivalent slot in the hypothesis (R10_HYPO_EQNS). The measures of hypothesis acceptability and tractability are set to indicate the hypothesis is so far acceptable even though it has only been generalized from one model; later (dis)confirmatory testing will find out if the hypothesis is really acceptable. Four hypotheses are constructed by STERN in this

manner (III, Table 4.1).

Since STERN has now found hypotheses using the strategy for the generalization of experimental results into hypotheses, control returns to the top level in STERN (*Strategy chooser* rules). New hypotheses have been found: the qualitative ones are applicable to all available experimental paradigms but the quantitative ones are not. Thus STERN chooses to try the confirmation strategy (described in detail in Chapter 5) on the quantitative hypotheses. For completeness, we will briefly consider what happens to the four new quantitative hypotheses during confirmation.

### 6.5.3 Partly Disconfirming The New Quantitative Hypotheses

The four hypotheses discovered by generalization are shown to be unacceptable by the confirmation process. The hypotheses are unable to account for the experimental paradigms that were not involved in their original formation. Attempts at making predictions fail when the hypothesis (law) describing the period of swing of a pendulum is applied to the inclined plane paradigm. This is because there is no parameter in that paradigm that corresponds to the *size* parameter (i.e. the length of the pendulum cord). STERN has thus correctly demonstrated that the pendulum law is just a model. Two of the other three hypotheses are variations of Galileo's law of free fall. STERN manages to generate predictions for the pendulum experimental paradigm with both hypotheses. However, their predictive instances do not match well with the experimental test results. The equations were originally inferred from the linear motion of the inclined plane. Thus STERN has correctly found that these equations cannot apply to the curved path of the pendulum. In other words the two variations of Galileo's law are not the most general form of the law.

## 6.6 STERN ASSESSMENT ON GENERALIZATION

### 6.6.1 Completeness Of The Generalization Strategy

Significant discoveries have been made by STERN using the strategy for the generalization of experimental results into hypotheses. All the terms that are relevant to the characterization of the phenomenon (i.e. speed, height, time and length), and those that are irrelevant (volume, weight), have been identified. Further, those terms that are relevant to only one experimental paradigm have also been delimited (such as the size and angle of the pendulum). Thus, STERN possess a full (mainly qualitative) understanding of the phenomenon. Quantitative models that describe the phenomenon in specific experimental paradigms have also been discovered (and shown by the confirmation strategy to be specifically limited to particular experimental paradigms).

Being able to deal with noisy data is something that computational models of scientific discovery should be able to do. The experimental simulator adds noise to instances but STERN is well able to cope with it. During the generalization of instances into quantitative models, noise in the data is naturally dealt with by the combination of: (i) the least squares method applied to the log-log "graph" of independent and dependent terms; and (ii) the accuracy of the match between the rational fraction and the index. Furthermore, this method allows STERN to find several correct model-equations without producing a false positive equation.

Designing experiments is a significant part of scientific discovery. STERN's design abilities are sophisticated. Of the many experimental tests designed by STERN, no experiments are produced that consider trivial relations between parameters or that are irrelevant to the phenomena being investigated. The use of background knowledge and pragmatic knowledge about experiments are essential components of this ability.

The extent of communication between the theoretical and experimental sides of the research programme is less than in the confirmation strategy. However, this

should not be considered a deficit. The reduced amount of communication simply reflects the fact that the design and performance of experiments, to build up a body of empirical results, does not require any theoretical considerations. Nevertheless, the correspondence between experimental parameters and theoretical terms has an underpinning role in the interpretation of experimental results into instances. It is only by the use of correspondence relations that STERN is able to determine that the law governing the period of pendulum swing was limited just to the pendulum paradigm.

In general, we have seen that STERN is well able to model the generalization from experimental results to hypotheses.

### 6.6.2 Advances On Previous Work

The generalization of empirical data into higher-level theoretical knowledge is the area that many previous scientific discovery systems have modelled. Typically, the current systems model the generalization of one set of data into a parsimonious description that constitutes a model (derived from several instances). Some of the computer models have considered quantitative inferences in this respect (e.g. BACON, Langley *et.al*, 1987; FARENHIET, Zykow, 1987), others have considered qualitative inferences (e.g. STAHL, Langley *et.al*, 1987; GELL-MANN, Fisher & Zytkow, forthcoming). Two program have even combined both types of inferences (e.g. IDS, Nordhausen & Langley, 1987; ABACUS, Falkenhainer & Michalski, 1986).

The generalization strategy in STERN not only models the generalization of instances into models, but also considers: the design and performance of experiments; the interpretation of experimental results; and the generalization of models into hypotheses (for both quantitative and qualitative representations). STERN clearly covers more ground than previous programs, but it does more than that. Previous systems are in effect subsumed by STERN. For example, BACON.1

could in principle be substituted for the *Generalize instances* rules and not make any difference to STERN's overall performance.

BACON (Langley *et.al*, 1987) and ABACUS (Falkenhainer & Michalski, 1986) use regularity spotters in the inference to equations from numerical data. To discover power relations STERN uses the more efficient log-log "graph" technique described in §6.5.1. However, STERN also uses regularity spotters in the shape of its qualforms but they differ in many ways from "conventional" regularity spotters. First, STERN's qualforms constitute explicit declarative knowledge in the program. Second, the number of different qualforms in STERN is more than double that of "conventional" regularity spotters. Third, the qualforms are considered at all levels of theory. These advances on the use of regularity spotters underlie many of STERN's powerful qualitative reasoning abilities. We will see how these abilities extend beyond the generalization to the generation of new quantitative hypotheses from old (Chapter 7).

STERN's qualforms are like the qualitative schema of IDS (Nordhausen & Langley, 1987) but are somewhat simpler. STERN and IDS differ in that IDS's qualitative schemata attempt to characterize the behaviour of a phenomenon as it progresses through a number of discontinuous states (e.g. melting, boiling). The schemata are induced as a preliminary step and form the basis for quantitative reasoning. STERN also uses its qualforms to infer quantitative theoretical knowledge but also employs them in other ways; for example to test the validity of substitutions of terms when confirming hypotheses (Chapter 5).

Previous models that have had some ability to design experiments have done so when some theoretical prediction is to be tested (i.e. Rajamoney *et.al.*, 1985; Kulkarni & Simon, 1988). STERN can do the same (as we saw in Chapter 5) but it can also design and perform a comprehensive range of experimental tests without reference to theory. This depends on the rich representation of experiments STERN possesses and the ability to use background knowledge.

In terms of Klahr & Dunbar's (1988) SDDS model one pattern of discovery is similar to STERN's generalization strategy (see Figure 2.2). The process covers the *induction* of a *frame* by the *generation* of experimental *outcomes*. This involves *focusing on*, *choosing* and *setting* the experimental variables to permit a trial to be *run* and outcomes to be *observed*. The outcomes are *generalized* into a frame. This SDDS process is equivalent to the design and performance of an experimental test and the generalization of the results into a model. Thus STERN subsumes SDDS as well as partly implementing its processes hierarchy.

### 6.6.3 Conclusions

In this chapter we have seen how STERN is able to obtain experimental results and generalize them into hypotheses. STERN also overcomes the limitations of previous discovery systems that generalise data into models. In particular, STERN:

- Designs experiments in the absence of (and also with) any theoretical predictions using background knowledge and the pragmatic information on experiments.
- Interprets experimental results into theoretical inferences (STERN does not simply assume that experimental results are true data that feed straight into theoretical inferences).
- Employs a wide range of explicitly represented regularity spotters (i.e. qualforms), on all the levels of theoretical knowledge, that are used in many diverse ways throughout the program.
- Takes into account noise in experimental data during its generalization of instances into quantitative models.
- Obtains a deep and broad understanding of a domain by finding a wide range of qualitative hypotheses that indicate which terms are relevant or irrelevant for the characterization of the phenomena.
- Avoids an explosion in numbers of models that apply to experimental setups by generalizing models into hypotheses that apply to experimental paradigms.

STERN can perform the generalization of experimental results into theories. In

the modelling of the Galilean domain many qualitative hypotheses were formed but no successful quantitative hypotheses. In the next chapter we will see how STERN finds successful quantitative hypotheses from the acceptable qualitative hypotheses and the unacceptable quantitative hypotheses discovered here. In particular, we will see how STERN infers the most general form of the law of free fall.

# Chapter 7

# New Hypotheses From Old

## 7.1 INTRODUCTION

Galileo did not find the law of free fall by a direct process of inductive generalization from quantitative experimental data. We have seen in previous chapters how the Aristotelian laws were rejected and how experiments were performed to gather more information about the phenomenon in question, using two different strategies. Thus a wealth of information about the phenomenon is available, including: some definitely incorrect hypotheses, others that are only applicable in specific circumstances, and many generally-acceptable qualitative hypotheses. It was from this extensive body of knowledge that the law of free fall was eventually proposed. The law of free fall was finally accepted by Galileo when all of the proposed hypotheses had been tested against experimental results.

STERN models this important aspect of scientific discovery using its strategy that infers new hypotheses from old. The type of new hypotheses that STERN seeks are quantitative ones; that is ones with equations. To generate new hypotheses from old there must be some existing hypotheses. STERN chooses the strategy when attempts have been made to test the existing hypotheses against all the manufactured experimental paradigms. The strategy is instantiated in STERN's *New Hypotheses* rule class (Table 7.1). STERN uses the strategy to find not only the correct law of free fall but many other quantitative hypotheses (although only the free fall hypothesis will eventually be shown to be acceptable). This chapter considers the details of how STERN generates new hypotheses from old. We will first consider the theoretical basis for generating new hypotheses and then see how STERN actually finds new hypotheses. Finally, we will consider how STERN

## Table 7.1 NEW HYPOTHESES Rules (RULES_14)

**R14_QUAL_TO_EQNS**

**Condition:**

All the hypotheses in the research programme have been tested experimentally.

**Action:**

A multi-stage procedure with four processes to specifically find new equations:

(1) from all adequate qualforms find the set of pairs of increase and from_zero qualforms that have the same terms; (2) eliminate from the set any pairs of qualforms that have terms that are referred to by a semi acceptable quantitative hypothesis; (3) isolate the terms from the qualforms and generate exponential equations from the pairs of terms, with the term last in each of the pairs of qualforms being the term on which the exponential function operates, and the index ranging over all combinations of fractions that have denominator and numerator number equal to or less than 3;

and for each equation generated construct a new hypothesis with the group slot filled by a symbol that relates all the equations that were originally inferred from the same pair of terms.

overcomes the limitations of previous discovery systems which perform similar discovery tasks.

Before we considered the specific way in which STERN generates new hypotheses, we should note that STERN's approach is just one of several different methods that exist. For example, scientist may generate new hypotheses by analogy to theories found in related research programmes or even in quite different fields of science. Investigations of alternative approaches will be interesting work for the future that will build upon STERN's present abilities.

## 7.2 GENERATING NEW HYPOTHESES FROM OLD

Before we consider how STERN actually generates new hypotheses from old, we need to understand why STERN employs the techniques it does. Existing hypotheses can be used in three ways in the inference to new hypotheses. Known hypotheses can: (i) indicate which theoretical terms are relevant and irrelevant to the characterization of the domain; (ii) rule out specific forms of equations; and (iii) suggest likely forms of potentially acceptable equations.

### 7.2.1 Relevant And Irrelevant Terms From Qualforms

The strategy that STERN uses to generalize experimental results into hypotheses involves the discovery of many acceptable hypotheses with qualforms. STERN uses many different qualforms (Table 4.13). Qualforms state qualitative relationships that have been found between two terms; for example, (INCREASE A B), indicates that the magnitude of *A* increases monotonically with *B*. Qualforms in acceptable hypotheses thus contain valuable information that is useful in the present task. One way the qualforms can be used is in determining which terms are relevant or irrelevant for characterizing the phenomena.

Consider a qualform with a STEADY predicate; for example (STEADY T_V T_W). This qualform indicates that T_V is constant as T_W varies. Thus, T_W is not

functionally related to T_V. When the qualform belongs to an acceptable hypothesis this means that the qualform has been found to apply in a way generally applicable to the phenomena. Therefore, we can conclude more generally that the phenomena will not be characterized by any equation that relates T_V and T_W. Similarly, an INDEPENDENT qualform indicates that there is no known relation between the two terms, thus its terms would not appear in an acceptable equation either.

Furthermore, most of the qualforms indicate definite relations between terms (e.g. (INCREASE T_V T_H)). Thus any term that does not appear in any qualforms whatsoever is unlikely to be relevant. For example, the term for the length of the chord in a pendulum, T_S, does not appear in any of the acceptable qualforms that STERN has found (Table 4.1). This is hardly surprising as this term only refers to a parameter that occurs in the pendulum paradigm.

Relevant terms can also be found using qualforms from acceptable hypotheses. For example, (INCREASE T_V T_H) indicates that T_V and T_H are in a functional relationship. Thus we would expect an equation to include both terms. The same argument applies to all the other qualforms except those with the STEADY or INDEPENDENT predicates (Table 4.13).

By a combination of all three techniques just described, all the terms in a research programme can be classified as relevant or irrelevant. For example, STERN finds that T_TIME, T_V, T_L and T_H are relevant when modelling the Galilean episode. The irrelevant terms found include T_S, T_VOL and T_W, amongst others.

### 7.2.2 Unlikely Terms And The Forms Of Equations

STERN has existing hypotheses with equations in addition to hypotheses with qualforms. None of these equations are acceptable. Some are unacceptable and some partly acceptable. However, it is because they are not completely acceptable that they can play a part in the generation of new hypotheses in one of two ways, depending on their degree of acceptability.

First, an equation that is partially acceptable indicates that its terms cannot form

the basis of a generally acceptable equation. Consider an hypothesis with an equation that is only partly acceptable; for example:

$$(= T\_D\ (*\ T\_TIME\ T\_TIME)), \quad \ldots (7.1)$$

This is an expression of the law of free fall that is only applicable to the inclined plane experiment. From this equation we can deduce that the generally acceptable hypotheses will not include just T_D and T_TIME. On the one hand Equation 7.1 only applies to the inclined plane experimental paradigm, so it is not generally acceptable; and on the other hand, any other equation in terms of T_D and T_TIME alone cannot apply to the inclined plane paradigm, so again is not acceptable. Therefore, no equation in T_D and T_TIME will be acceptable. In general, a partly acceptable hypothesis cuts down the range of potential new hypotheses dramatically with respect to the terms that are considered in its equation.

Second, a completely unacceptable hypothesis helps to cut down the space of new equations to be considered by explicitly ruling out one equation. For example, the two Aristotelian effective weight laws were found to be unacceptable by the disconfirmation processes. Thus, the processes to generate new hypotheses need not bother to consider them during its processing. An unacceptable quantitative hypothesis only indicates that a particular equation does not account for any experiments. The modification of the equation may in fact produce a generally-acceptable hypothesis. For example, it cannot be inferred from the unacceptable Aristotelian effective weight laws that no other equation in those terms could be an acceptable hypothesis. However, in this case, a new acceptable equation is unlikely, because the effective weight and density terms do not appear in any acceptable qualforms (see §7.2.1).

### 7.2.3 Suggesting The Form Of Equations

To recap, there are inferences that can be made which will find likely combinations of terms as candidates for new hypotheses and qualitative formalisms have significant role in this. However, qualforms are also important when it comes

to postulating the structure of equations that may relate previously-selected terms. Because qualforms indicate that certain regularities obtain, equations which satisfy those regularities are likely to be good candidates.

Given the two acceptable hypotheses with the qualforms (INCREASE A B) and (INCREASE C B) a likely form that an equation involving *A* and *C* may take is:

$$f(A) = g(C) , \qquad \ldots (7.2)$$

where *f* and *g* are monotonic functions of *A* and *C*. The rationale behind this inference is that as both *A* and *C* increase in magnitude with respect to the term *B*, so *A* and *C* may themselves be directly related.

Conservation laws are often favoured by physical scientists (Feynman, 1965). So combinations of qualforms that suggest that two terms may be combined to yield a constant quantity are worth considering. For example, consider the pair of qualforms (REPEAT+ A B) and (REPEAT- C B). The first one states that as *B* increases monotonically, *A* increases from an initial value, rises to a maximum value and returns to the initial value. Similarly with the second qualform, except that *C* decreases to a minimum value. Reasonable equations that fit this pair of qualforms are:

$$f(A) + g(C) = \text{const} , \qquad \ldots (7.3a)$$

$$f(A) \,.\, g(C) = \text{const} , \qquad \ldots (7.3b)$$

where the *f* and *g* are again simple monotonic functions and *const* is some arbitrary constant. Other combinations of qualforms may also imply conservation equations. For example, two qualforms of the same type [e.g. (increase A B) and (increase C B)] may be satisfied by equations 7.3a and b with their operators replaced by '-' and '÷', respectively.

Clearly, only the general form of the equation is implied by a combination of qualforms so the potential range of one type of equation is infinite; there is no limit to the range of form that the monotonic functions *f* and *g* may take. However,

only a small proportion would be considered by a scientist, for example exponential relationships with rational indices under 3. Search methods like those embodied in BACON.6 (Langley *et.al.*, 1986) might also be employed.

Thus we have seen the different methods that will form the basis of STERN's ability to generate new hypotheses from old. In the next section the particular manner in which they are used in STERN will be discussed.

## 7.3 STERN'S NEW HYPOTHESES MECHANISM

### 7.3.1 Exponential Equations

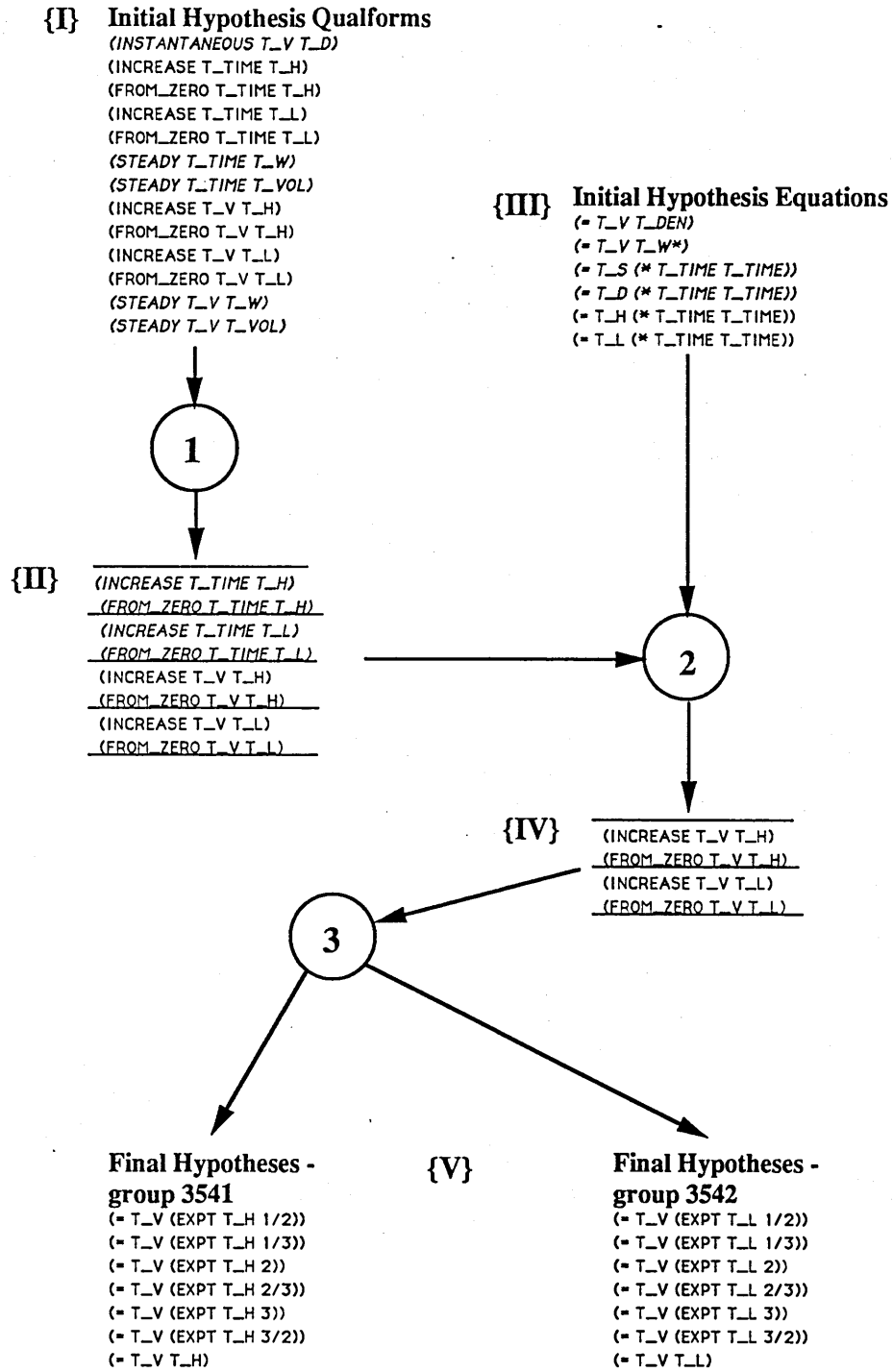
STERN models the generation of new hypotheses from old using some of the the techniques discussed (QUAL_TO_EQNS in *New Hypotheses*, Table 7.1). The equations that STERN finds are exponential equations with small rational indices. Such equations have two properties that are particularly relevant here: (i) they are monotonically increasing functions; and (ii) they pass through the origin. STERN has qualforms that can identify when the values of two terms have these properties; namely the INCREASE and FROM_ZERO qualforms. To be specific; (FROM_ZERO A B) means that *A* and *B* pass though the origin and (INCREASE A B) indicates that *A* increases as *B* increases. Thus both can be considered as necessary conditions for an exponential equation of the form

$$(= (\text{expt B n}) \text{ A}), \qquad \ldots (7.4).$$

where ***n*** is the index of the function. The next subsection describes in detail how STERN goes about searching for such equations.

### 7.3.2 Inferring Equations

The generation of new hypotheses in STERN is a four stage process (see Table 7.1 & Figure 7.1). First, existing qualitative hypotheses are analysed for pairs of INCREASE and FROM_ZERO qualforms that have identical terms. Four such pairs of qualforms are found. Second, those pairs of qualforms with terms referred to in partially-acceptable hypotheses are eliminated, leaving only two pairs. Third, the terms themselves are isolated and exponential equations are generated for each pair




## Figure 7.1 New Hypotheses From Old

*Italics* indicates qualfroms that will be eliminated or will be ignored by the next process.
See Table 7.1 for a description of numbered processes

of terms according to equation 7.4. The values taken by the index $n$ are all the different combinations of fractions, $p/q$, given by numerator and denominator integers up to a user defined limit. The limit typically chosen is 3; Galileo never consider power equations in which $p$ or $q$ were greater than 3 and very few laws of nature have values of $p$ and $q$ above 3 (when they can be expressed as power equations with rational indices). As each pair of terms yields more than one equation, the several hypotheses generated are given a unique symbol indicating that they belong to particular set inferred by the same process. In Chapter 6, we saw how this information was used to cut down significantly the number of hypotheses considered, once one had been found to be acceptable.

Fourteen hypotheses originating from two pairs of terms were in fact generated. The hypothesis with the equation

$$(= \text{T\_V } (\text{EXPT T\_H } 1/2)), \qquad \ldots (7.5)$$

is the correct law of free fall, although at this stage any hypothesis could be acceptable. When the generation of a new hypothesis is complete, control is returned to the top level (*Strategy chooser*). STERN now chooses the confirmation strategy to attempt to test the new hypotheses against the available experiments. However, as the Aristotelian law is no longer believed to be true, the process fails to generate models from the hypotheses. The necessary constant speed condition no longer applies, so speed term (T_V) cannot be substituted by its definition. Thus, the tractability of the hypotheses is in effect reduced whilst their acceptability remains to be considered. To test the new hypotheses the invention of new experimental paradigms is required, as we will see later (in Chapter 8).

### 7.3.3 Summary

The formation of new hypothesis from old ones can be conceived of as a serial search of two spaces, guided by the existing hypotheses and their various degrees of acceptability. The first search is in the space of all the theoretical terms in the research programme. This search finds the terms that are relevant to the

characterization of the phenomenon in all the experimental paradigms of the research programme. The second search is through the space of general forms of equations for ones that are compatible with accepted findings. The first and second processes in Figure 7.1, perform the search of the space of terms in STERN, and the third process attempts to find equations with suitable forms.

The processes taken together can also be considered as one of specialization, modifying qualforms into equations that cover a very much more restricted set of interrelations between the terms.

## 7.4 STERN ASSESSMENT ON NEW HYPOTHESES

### 7.4.1 Completeness

From the nineteen hypotheses obtained from the previous discovery stages, STERN was able to generate 14 new hypotheses in two sets, one of them containing the correct law of free fall. The techniques used are a powerful means of finding potentially acceptable hypotheses that may be generally applicable. The power of the processes resides in the fact that all previous hypotheses are considered. Relevant and irrelevant terms are found from the qualitative hypotheses and partially acceptable quantitative hypotheses. Likely forms of equations are found using the acceptable qualitative hypotheses and unacceptable quantitative hypotheses.

Without this strategy to generate new hypotheses from old, STERN would not have been able to find the true law of free fall. Furthermore, in using this method to find the law, STERN is closely modelling the way in which Galileo found the same law. Galileo used the knowledge which he had gained from performing experiments to postulate possible forms of a general law of motion (See Chapter 3).

### 7.4.2 Advances On Previous Work

The processes to generate new quantitative hypotheses described above paid

particular attention to the degree of acceptability of all existing hypotheses. STERN relies on the extent to which hypotheses have been applied successfully to all the experimental paradigms in the research programme. No previous model of scientific discovery has considered the inference to new theoretical knowledge using both qualitative and quantitative knowledge of such varying degrees of acceptability. However, some previous systems have employed techniques that are interesting to compare to STERN's abilities.

First, let us consider programs that modify one unacceptable model (or hypothesis) to form a model that is potentially more acceptable. A range of different methods are used in different programs. REVOLVER (Rose & Langley, 1986; Rose, 1988) revises inconsistent chemical reactions by adding or subtracting atoms from premises about the structure of molecules. Various item of information are stored about the types of inference made and when they were made (*reduced lists & sources tags*). These items are used by REVOLVER in a function that evaluates how best to revise the reactions. Amongst KEKEDA's (Kulkarni & Simon, 1988) *proposition generators* and *proposition modifiers* are heuristics that suggest how reaction equations may be changed when unexpected experimental outcomes are obtained. The *evoke frame* and *use prior knowledge* processes in Klahr & Dunbar's (1988) SDDS process hierarchy are possible locations for the strategy considered above. Thus, a range of methods to modify existing unacceptable theoretical knowledge are used by previous systems. However, unlike STERN which takes into account all known hypotheses to infer a new one, the previous systems only attempt to modify individual laws that have been found to be inconsistent or inadequate. The power of STERN's new hypothesis generation strategy comes from its use of all available hypotheses.

Several programs have made use of regularity spotters. These roughly resemble STERN's qualforms in the way they function. BACON (Langley, *et.al.*, 1987) employs regularity spotters in its heuristics that look for simple *increase*, *decrease*,

*linear* and *constant* relationships between terms. When a regularity is found it is not explicitly represented but immediately used to define a new term. The regularity spotters in ABACUS (Falkenhainer & Michalski, 1986) are used for a similar purpose, but in a more sophisticated manner. ABACUS attempts to find all the relevant variables by applying regularity spotters to all the different combinations of pairs of variables (to find the maximal cycles in the proportionality graph) before making any quantitative inferences. The result is similar to STERN's search though its qualforms for relevant terms, but in ABACUS the search is data-led and only two types of substantive qualitative relation are considered. STERN can use nine qualforms so is able to find a much great variety of relations amongst terms. Furthermore, STERN qualforms are used in other inference processes throughout the program (Chapters 5 & 6). Qualitative knowledge about processes is stored explicitly in IDS (Nordhausen & Langley, 1987) but is only used in a most rudimentary way to infer quantitative laws. The main differences between STERN and previous systems in this respect is that STERN: (i) explicitly represents the qualitative relations it finds; (ii) assesses their acceptability across different experimental paradigms; and (iii) then uses them to find new quantitative hypotheses in a theory-led fashion. Other systems have tended to only use such knowledge implicitly in the generalization from observations to descriptive equations in a data-led manner.

### 7.4.3 Conclusions

In this chapter we have seen how STERN can generate new hypotheses from its existing ones. One of the laws found is the correct law of free fall. STERN's powerful strategy for generating new hypotheses encompasses many abilities. These include:

- Using acceptable qualitative hypotheses, and unacceptable and partially acceptable quantitative hypotheses, to distinguish terms that are either relevant or irrelevant for the characterization of the phenomena.

- Proposing the form of likely equations based on acceptable qualitative hypotheses and unacceptable quantitative hypotheses.
- The grouping together of mutually exclusive hypotheses generated from a single process, for future reference.

Although many new hypotheses have been generated, STERN has not managed to find any which are acceptable. In the next Chapter we will see how STERN has to invent new experiments to be able to test the new hypotheses successfully.

# Chapter 8

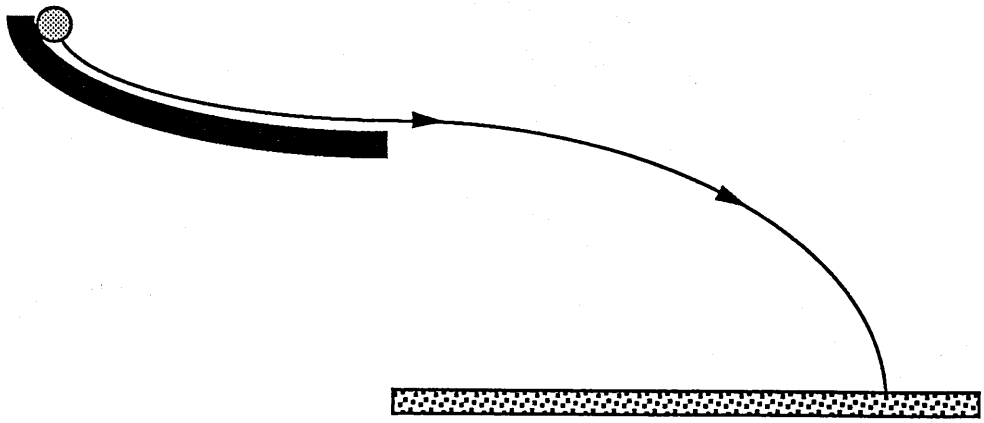
# Inventing New Experiments

## 8.1 INTRODUCTION

We have already seen how experiments have played a large part in the modelling of the discoveries in the Galilean domain by STERN. Experiments have been used to (dis)confirm hypotheses by allowing experimental test results and predictions to be compared. Experiments have also been designed and performed to obtain a large body of results that have been generalized into hypotheses. Now, in this chapter we will consider some quite radical things: the manufacture and use of new experiments; and the high level control of the availability of experimental paradigms.

Galileo invented new experiments by combining the experiments he already knew (see Chapter 3). For example, he realized that the inclined plane could be used as a launcher for the projectile experiments. The initial part of the combined experiment (inclined plane) allows the terminal part (projectile) to be investigated for the first time as the terminal part cannot be used in isolation. Galileo invented other combined experiments, such as the pendulum with shortening cord and the combined curved ramp and projectile (see Figure 8.1). Inventing new experiments by combing existing experimental paradigms is ubiquitous in science. For example, Newton used a double pendulum experiment (Figure 8.2) in which the two pendulums with bobs of different mass were placed side by side (Magie, 1935, 41). The initial pendulum is released and swings down to collided with the terminal pendulum. This experiment is one that Newton employed in the discovery of the conservation of momentum. In fact, it depends on a knowledge Galileo's law of free fall to determined the relative speeds of the pendulum bobs just before and after

(A) Curved Ramp And Projectile



(B) Pendulum With Shortening Cord

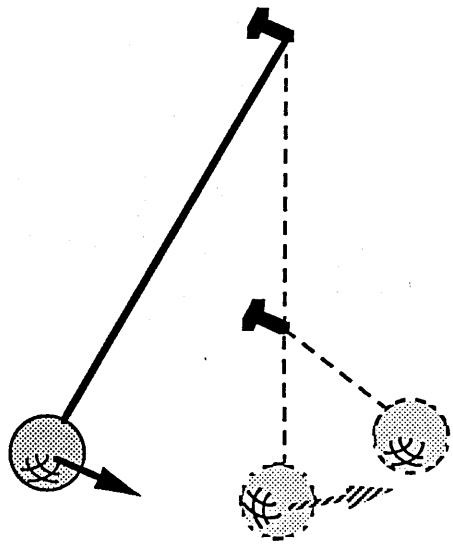


Figure 8.1 Two Of Galileo's Combined Experiments

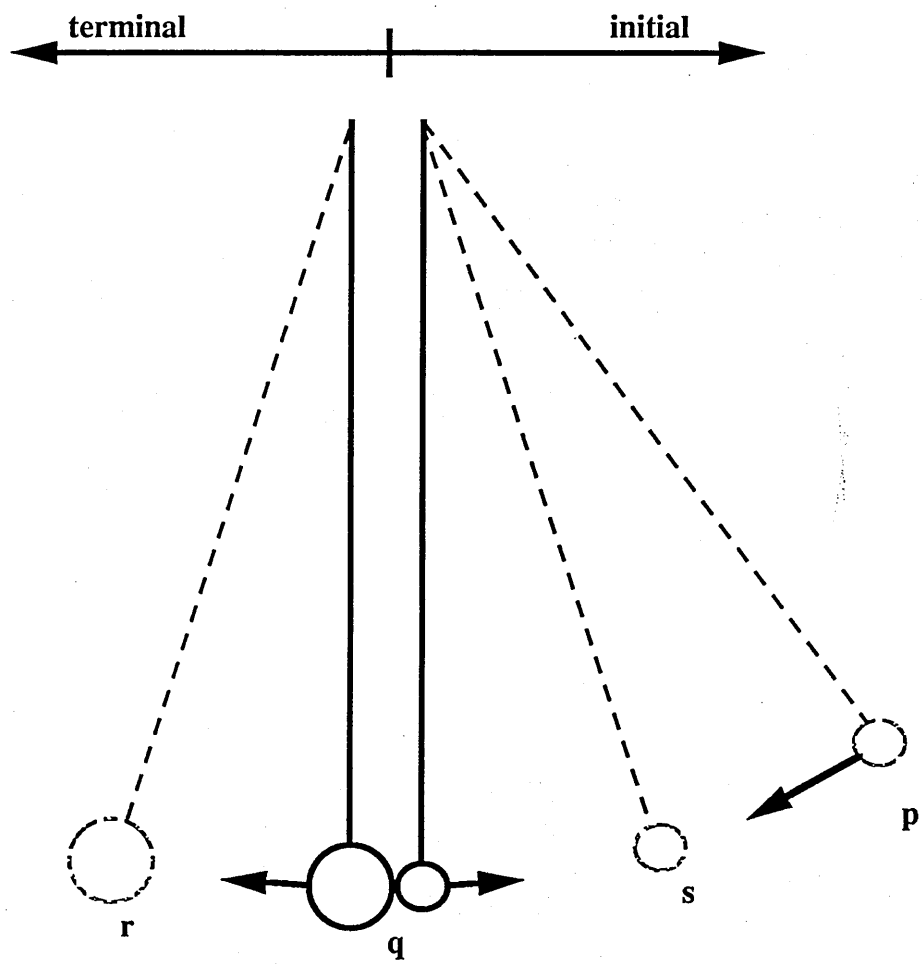



**Figure 8.2 Newton's Double Pendulum Experiment**

Key: *p* - release of initial pendulum, *q* - collision,
*r* & *s* - final positions.
Adapted from Magie (1935).

their collision. This is an example of how Newton's discoveries were built on the foundation provided by Galileo. More recently, subatomic physics experiments, like those performed at CERN, can be considered as combined experiments. The particle generator and accelerator constitutes the initial part that generates high energy particles. The bubble chamber in which the particles are smashed together is the terminal part. Although combining experimental paradigms is an important and widely used method to obtain new experiments it is not the only means that exists. For example, new experiments may be devised by examining paradigms used in other domains and adapting them, by analogy, to one's own research programme. Often new types of experiments come into being by the invention of new technologies. However, here we will just consider the combination process to devise new experiments; and in particular, the combination of just two experimental paradigms.

Inventing new experiments provide new variations of experimental parameters and conditions which can circumvent the limitations of previous experiments. Furthermore, new experiments may also allow solutions to be found for problems that have occurred during theoretical inferences. Galileo used his combined inclined plane and projectile experiment to do exactly that. Galileo had found his law of free fall but was only able to test it once the combined inclined planed and projectile experiment had been invented (we will see exactly why below).

Inventing new experimental paradigms had a crucial role in Galileo's discoveries. Thus, the modelling of this episode would be incomplete without some consideration of this important aspect. Fortunately STERN can cope with new experimental paradigms and setups. STERN does not devise new experimental paradigms just because it runs out of experiments; it is given many paradigms as input and they are not exhausted before STERN needs to consider new ones. STERN constructs new experimental paradigms to overcome the intractability of

certain hypotheses in theoretical inferences. Devising experiments involves *combining* known experimental paradigms; for example, the use of the inclined plane in conjunction with projectiles (see Figure 3.2).

Galileo knew of many simple experiments that he could perform and had the means to invent many new experiments. Clearly he did not manufacture and perform experimental tests on all the experiments he ever conceived. It seems that Galileo only used a relatively small number of experiments, in the main preferring pendulums and the inclined plane. STERN also considers a limited number of experiments at a time. The limiting of the availability of experimental paradigms in STERN has been found to be important to how efficiently STERN is able to make discoveries. Simply stated, if there are too many experimental paradigms the discovery process becomes to cumbersome. Hence, a mechanism is used by STERN to limit the availability of experiments.

In this chapter we will consider why STERN needs to consider new experiments and how it finds new combined experiments. We will also consider in detail why STERN needs to limit the availability of experimental paradigms and its mechanism for doing so.

## 8.2 NEW EXPERIMENTAL PARADIGMS & SETUPS

### 8.2.1 Why STERN Employs New Experiments

The are several reasons why STERN constructs new experimental paradigms. The first and the most important one is that new experiments make intractable hypotheses tractable; for example free fall hypotheses. These hypotheses are intractable for theoretical inferences because the speed term (T_V) cannot be eliminated when modelling either the pendulum or inclined plane experiments. In the confirmation strategy, STERN can only replace a term by its definition when certain specific conditions apply. To eliminate the speed term, speed must be constant. Previously, speed could be replaced because the Aristotelian law of

instantaneous acceleration was applicable, but it has by now been disconfirmed. Thus STERN is left with an intractable hypothesis. Considering new experiments may allow the speed term to be eliminated. For example, the free fall hypothesis can be applied to an inclined plane to find an expression for the speed down the plane. A second expression for the horizontal speed of the projectile can also be inferred. One expression can thus be substituted into the other to eliminate the speed term, leaving a single equation that just has measurable terms (see §5.3.1.3).

Second, new experiments are sometimes needed because certain experiments cannot be carried out in isolation. For example, experiments on the motion of projectiles cannot be considered without some means to launch the body into the air. The inclined plane is one suitable candidate for the role of launcher. In general, such an arrangement is one in which the *initial* part of the experiment (inclined plane) instantiates the phenomenon so that it can be observed in the *terminal* part (projectile). This is a technique widely used in experimental science.

The third and final reason why STERN might consider new experiments is if runs out of experiments to use. However, this eventuality does not arise when modelling the Galilean episode.

### 8.2.2 How STERN Constructs New Experiments

The basic strategy that devises new experiments in STERN involves combining two old experimental paradigms in to one new paradigm. This models what Galileo did, at least twice, in his investigations of naturally accelerated motion.

The most important new experimental paradigm devised by STERN is the combined inclined plane and projectile experiment (see Figure 3.2). A combined experiment has two parts. The *initial* part (the inclined plane) acts as a feeder into the *terminal* part (the projectile). Some means may be required to modify the exact behaviour of the phenomenon during the transition from the initial to terminal parts (e.g. the lip at the end of the inclined plane to convert the ball's angled descent into purely horizontal motion just as it becomes the projectile). In domains like

Galileo's, human scientists can see whether two experiments can be combined. How such inferences are made using real world knowledge is an interesting topic, but is beyond the scope of the present work. Thus, STERN is told as one of its inputs which paradigms can be legally combined. What is of particular interest here are the circumstances that lead to the invention of new experiments and how they are subsequently employed. Experiments may be performed on combined experiments in one of two *modes* that depend on whether the input-m parameter is in the initial or terminal part. As the names of the modes suggest, the input-m parameter is in the initial part in the *initial mode*, and in the terminal part in the *terminal mode*. In STERN, devising new combined experimental paradigms is a three stage process (*new paradigm* class of rules, Table 8.1).

In the first stage a single experimental paradigm is chosen to be the terminal part (R13_CHOOSE_TERMINAL). STERN prefers available (already manufactured) paradigms over ones that have only just been conceived and also chooses the paradigm with the most setups. The chosen paradigm, say the projectile paradigm, is made active.

To begin the second stage, initial parts for the terminal projectile are sought (R13_MAKE_COMBINES). STERN knows which paradigms have setups that are suitable initial parts for the projectile paradigm (they are named in the *combine* slot of its frame). For example, *down_incplane* is the name of one of the inclined plane's setups that is a suitable initial part. The construction of the combined experimental paradigms from the inclined plane and the projectile paradigms is a matter of instantiating a new paradigm frame and filling the slots that relate to each part, using the information available for the existing paradigms (see, Table 4.8). For example, the ease of manufacture of the new combined experimental paradigm is calculated from the values of its two parts. This measure is always a value between zero and unity; the larger the value the easier the paradigm is to

## Table 8.1 NEW PARADIGMS Rules (RULES_13)

**R13_COMBINED_SETUPS***
**Condition:**
There is an active list of experimental paradigms.
**Action:**
For each active experimental paradigm in the list, construct experimental setups from the paradigm;
the name of the setup is made by concatenating the initial and terminal setup names with a '+' between.

**R13_MAKE_COMBINES**
**Condition:**
There is a single active experimental paradigm,
and other experimental paradigms exist in the experimental side of the research programme.
**Action:**
The active experimental paradigm that is considered as a terminal part of a combined experiment;
search for suitable initial paradigms from amongst those in the research programme by examining the *combine* slot of the terminal paradigm's setup frames;
construct a combined experimental paradigm for each suitable initial paradigm with the active terminal paradigm, filling the slot of the paradigm frame from the two existing paradigms;
and calculate the ease of manufacture using the formula (i.t)/(i+t), where i and t are the values of manufacturing ease of the initial and terminal paradigms, respectively.

**R13_CHOOSE_TERMINAL**
**Condition:**
There is no active experimental paradigm,
and there are hypotheses stored in the theoretical part of the research programme,
and no combined experiments have been made under the active paradigm before.
**Action:**
Make active the experimental paradigm that has the greatest number of experimental setups that can act as terminal parts in a combined experiment, preferring those that have been manufactured over those that are just conceived, but not ones that are in combined experiments.

**R13_REDUCE_*_THRESHOLD**
**Condition:**
There is no active experimental paradigm,
and there are paradigms in the research programme that have not been "manufactured".
**Action:**
Choose the experimental paradigm with the greatest product of manufacture ease and number of experimental setups that has setups that are not just a terminal parts of combined experiments, and reduce the ease*setup parameter to a value as if the paradigm had been manufactured, thus making it available.

*The order of rules indicates their relative priority in conflict resolution.

manufacture. STERN calculates the value using the equation:

$$\text{manufacture ease} = (i \,.\, t) \,/\, (i + t)\,, \qquad \ldots (8.1)$$

where $i$ and $t$ are the ease values for the chosen initial and terminal paradigms. Equation 8.1 always yields a value between 0 and 1, that is less than the magnitude of either $i$ or $t$ alone. The name of the new experiment is a concatenation of the names of its parts (e.g. 'incplane+projectile').

The third and final stage of the process involves making setups for the combined experimental paradigm (R13_COMBINED_SETUPS). This involves constructing new experimental setups using the information stored in the new paradigm frame and storing the setups under the paradigm. STERN interrogates the terminal paradigm to find which combinations of setups are legitimate to combine (i.e the contents of the *combine* slot). The names of setups are concatenations like those of the paradigms.

STERN repeats all three stages of the process for all the experimental paradigms it knows. In addition to the combined inclined plane and projectile, STERN constructs five other new paradigms, such as the combined curved ramp and projectile, and a double pendulum with a shortening cord (See Figure 8.1). All the experimental paradigms are considered by STERN, when it decides to employ new experiments, because the process that controls the number of available experiments needs detailed information about the new experimental paradigms; as we will now see.

## 8.3 CONTROLLING THE AVAILABILITY OF EXPERIMENTS

### 8.3.1 Why Limit The Numbers Of Experiments?

STERN is given six experimental paradigms as its initial input and we have seen how it makes six more combined experiments. To cope with such a profusion of experiments STERN uses a mechanism to limit the number of available paradigms. There are three reasons why STERN needs to limit the availability of experimental paradigms. First, as Galison (1986) notes, scientists working in a particular field

will not use all the experimental paradigms that are present in the field. They have preferences for particular experiments. Galileo seemed to concentrate on pendulums, inclined planes and projectiles, whilst other researchers favoured curved ramps.

Second, considering only a limited number experiments at a given time is a heuristic that improves STERN's efficiency. For example, attempting to confirm hypotheses using just two experimental paradigms saves considerable effort. Those hypotheses that are unacceptable can be been found using just two paradigms and so eliminated from further investigation. Thus effort is only expended on the testing of the remaining (partially) acceptable hypotheses with additional paradigms. A similar argument applies to the generalizing of experimental results into hypotheses. The minimum number of experiments that is necessary to base the generalization on is two, if there is to be a reasonable chance that the inferred hypothesis is general. Therefore, STERN only bothers to make generalizations from two sets of experimental results rather than wasting effort on more.

Third, the mechanism differentiates between experimental paradigms that are available for STERN to use and those that are not. This models the real difference that exists in science between experiments that have actually been physically constructed and those that have only been conceived.

### 8.3.2 Controlling Available Experiments In STERN

To limit the number of experiments that are available STERN calculates a pragmatic value for each experimental paradigm. This value is the product of (i) the measure of the ease of manufacture of experimental setups of the paradigm and (ii) the number of setups. STERN only considers those experimental paradigms that have a value of this product above a certain limit. Initially this limit is chosen (by the user) so that typically two experimental paradigms will be available. Less than two paradigms means that STERN cannot assess the acceptability of hypotheses or generalize experimental results to hypotheses. Using two experimental paradigms

also models Galileo's preference for his two favourite paradigms. Thus, STERN typically starts with the pendulum and inclined plane experimental paradigms as available experiments. Later when new experimental paradigms need to be made available, because the existing paradigms have been exhausted, the limit on the pragmatic value is lowered (by R13_REDUCE_*_THRESHOLD). STERN chooses the new value so that just one new paradigm becomes available.

Whilst modelling the Galilean episode, STERN decides to make a new experimental paradigm available just after new experimental paradigms have been constructed. The combined inclined plane and projectile experiment is made available and permits STERN to go on to confirm the free fall hypothesis.

## 8.4 STERN ASSESSMENT OF NEW EXPERIMENTS

### 8.4.1 Completeness

The construction and use of new experiments are important processes that STERN is able to model. STERN would not be such a complete discovery system if it did not have these abilities. Furthermore, STERN would have been unable successfully to model the Galilean domain, because there would have been no way to confirm that the law of free fall was correct. The main reason STERN considers experiments is to allow intractable hypotheses to become tractable.

STERN controls the availability of experimental paradigms. This enhances the performance of STERN by improving how efficiently it makes discoveries.

### 8.4.2 Comparison With And Advances On Previous Work

Previous discovery systems have not modelled the experiment component of scientific research programmes. This occurs even though the role of experiment has been acknowledged theoretically in Cognitive Science; for example in the cyclical account given by the BACON school (see Chapter 2). Typically, the only manifestation of experimentation is in the form of observational data that the

program is given. However, there are three limited exceptions.

HDD (Reimann, 1990), KEKEDA (Kulkarni & Simon, 1988) and Rajamoney et.al.'s (1985) programs all have some representation of experimental tests. Their representations consider the input-m and output parameters of the experiment, and the values that the parameters take. However, there is no consideration of pragmatic knowledge, such as which parameters are the easiest to manipulate and observe. Such knowledge is essential for the selection and design of experiments, particularly when no specific prediction is being tested. None of the systems consider the higher setup and paradigm levels of experimentation. This means that they are not able to to assess the general applicability of models or hypotheses.

As the representation of experimental knowledge is so poor, few processes that involve elements of experiments have been modelled by previous systems. The interpretation of experimental results to instances has not been considered as most previous models do not even distinguish between experimental tests and instances. The assessment of the accuracy of predictions is absent for the same reason. There are many different types of communication that occur between theory and experiments, but none have really been examined in the existing systems. COPER (Kokar, 1986) for example has units for its theoretical terms, but they are given as program inputs and are not used to refer to parameters in experiments.

The three programs that do have limited representations of experiment use them is different ways. In HDD, the representation of experimental tests act as a store for the pre-designed tests that are given as input. KEKEDA does design experimental tests, but only when there is a particular reaction to be tested and using mainly domain specific heuristics. Rajamoney *et.al.*'s (1985) program is also theory led and domain-specific. Incidentally, SDDS (Klahr & Dunbar, 1988) gives locations for experimental design and performance in its processes hierarchy.

The experimental representations and processes modelled by previous systems are limited compared to STERN. STERN is much more complete, it can: select

different experimental paradigms and setups using pragmatic experimental knowledge; design experimental tests in a general fashion using background knowledge, in the presence of a theory or not; construct and use new experimental paradigms from existing experiments; and control the availability of the experiments using pragmatic assessments of the experiments.

### 8.4.3 Conclusions

In this chapter we have seen how STERN constructs new experiments, by combining old experimental paradigms, and uses them to make further discoveries. STERN also controls the availability of experimental paradigms as a means to enhance the efficiency with which it makes discoveries. To summarize, the program's processes that deal with experiments allow STERN to:

- Construct new experiments by combining existing experimental paradigms.
- Employ new experiments to make further discoveries.
- Use combined paradigms as experimental solutions to intractability problems in theoretical inferences.
- Use combined paradigms to instantiate experiments that would otherwise have been impossible to perform in isolation.
- Control the availability of experimental paradigms as a way of enhancing the efficiency of the system, by cutting down wasted effort during the confirmation of, and the generalization to, hypotheses.
- Model the complex inter-play between the theory and experimental components of a research programme at a high level.

This ends our consideration of STERN experimental abilities, and it also draws to a close our considerations of STERN's many and varied processes that constitute its extensive range of powerful discovery abilities.

# Chapter 9

# Conclusions: The Cognitive Science Of Scientific Discovery

## 9.1 INTRODUCTION

One of the points we made at the beginning of this thesis was that our understanding of scientific discovery has not been reflexive. Science has advanced our knowledge of very many aspects of our universe in great detail. However, we do not have such a deep understanding of the nature of science and how scientists make discoveries. It is only recently that direct empirical investigations and theoretical studies have been carried out to further our knowledge in this area. Cognitive Science is the field in which much of this work has taken place. It is comprised loosely of three areas that have employed very different methods to investigate scientific discovery. Cognitive psychologists have performed empirical studies on simulated scientific discovery tasks. Philosophers of science have propounded many and varied theses of how science is performed, and even prescribed how it ought to be carried out. In Artificial Intelligence (AI) computational scientific discovery systems have been built to model episodes from the history of science.

This thesis is clearly located in the AI camp. However, the approach is somewhat different in that a framework for scientific discovery has been proposed. The framework proposes a minimum set of components that must be possessed by accounts and models of scientific discovery if they are to be acceptable (Chapter 1). The review of existing computational models and empirical studies was organized, and to some extent assessed, in terms of the framework (Chapter 2). STERN is a discovery system that models Galileo's discoveries on naturally accelerated motion

(Chapter 3). STERN instantiates all the components of the framework and has successfully made the same discoveries as Galileo (Chapters 4 to 8).

So, we have the field of cognitive science which investigates scientific discovery and a framework that partially characterizes scientific discovery. It is interesting to consider what we might learn by using the framework to analyse reflexively the investigation of scientific discovery in Cognitive Science. This is main objective of this chapter. We will consider:

*(§9.2) Mapping The Study Of Scientific Discovery Into The Framework.* There are many different approaches to research on scientific discovery in Cognitive Science. The different types of study, and the entities in them, neatly map onto components and items in the framework.

*(§9.3) The Experimental Component Of The Cognitive Science Of Scientific Discovery.* The framework's experimental component encompasses the empirical studies and the historical episodes considered in Cognitive Science.

*(§9.4) The Theoretical Component Of The Cognitive Science Of Scientific Discovery.* The framework's theoretical component encompasses the theories proposed by the researchers in all three areas and the computational models from AI in particular.

*(§9.5) The Acceptability Of Computational Models Of Scientific Discovery.* The theoretical work in cognitive science can be assessed according to criteria that are equivalent to those often used in other sciences. How well the existing computational models do according to these criteria is assessed.

We will also consider some other issues that are relevant to the computational modelling of scientific discovery; *Remaining Issues And Thoughts (§9.6).*

## 9.2 MAPPING THE STUDY OF SCIENTIFIC DISCOVERY INTO THE FRAMEWORK

The framework proposes a minimum set of components that seem to be essential for the characterization of scientific research programmes. In it, a *research programme* consists of a theoretical component and an experimental component that together investigate and characterize some delimited set of phenomena. Theoretical knowledge is viewed as *state transformation functions* and three types of theoretical knowledge are distinguished - *hypotheses*, *model* and *instances*. The acceptability of theoretical knowledge is assessed using acceptability criteria; for example explanatory breadth. Similarly, there are three levels of experiments - *experimental paradigms*, *setups* and *tests*. On some phenomena no experiments can be performed and only observations made; in these cases the reliability of experimental test results is substantially reduced. The framework also acknowledges that multiple types of communication occur between all the levels of the two main components. We will consider the experimental component, the theoretical component, and theory acceptability in the analysis of work in Cognitive Science. The Galilean episode will be used for comparison throughout (see Table 9.1).

### 9.2.1 The Experimental Component

Galileo's research programme was on the delimited phenomena of the motion of naturally accelerated bodies. On the experimental side Galileo used inclined planes, pendulums and invented new combined experiments; each of these is an experimental paradigm. A particular configuration of the inclined plane is an experimental setup. Performing a test on the setup is an experimental test that gives specific results, such as lists of values for input-m and output parameters. The comparison of the test result and a prediction relating to the inclined plane is an example of the communication that occurs between theory and experiment.

Now let us assume that the framework applies to the study of scientific discovery

**Table 9.1 Galileo Versus The Cognitive Science Of Scientific Discovery Under The Framework**

| Framework Components | Galileo on the motion of naturally accelerated bodies* | Cognitive Science of Scientific Discovery* |
|---|---|---|
| Research Programme | Naturally accelerated motion | Scientific discovery |
| Experiment | | |
| Experimental Paradigms | Inclined planes, pendulums, combined experiments | Computer simulated environments, specific tasks |
| Experimental Setups | (An inclined plane with fixed height) | (REFRACT, Wason's '2 4 6') |
| Experimental Tests | (Particular predictions on inclined plane) | Historical episodes of discovery & (results from REFRACT) |
| Activities | (Inventing combined experiments and running tests on them) | (Designing REFRACT & '2 4 6' task, and running tests on them) |
| Reliability | (Acknowledging the influence of noise) | ("Averaging" behaviour of many subjects) |
| Theory | | |
| Hypotheses | Aristotelian effective weight laws, law of free fall, etc. | Philosophical & psychological theories (BACON school cyclic account, the Framework) |
| Models | (Descriptions of motion on the inclined plane, with fixed height) | Computational models: Previous discovery, e.g. BACON; e.g. STERN |
| Instances | (Specific predictions on the inclined plane) | Runs of discovery systems on particular domains with specified inputs. |
| Inferences | (Make prediction for inclined plane with free fall law) | (Writing STERN using Framework, running STERN) |
| Acceptability Criteria | Success of applying hypotheses to experimental paradigms | Completeness, generality, internal coherence. |
| Communi-cation | (Comparing inclined plane prediction and test results) | Comparing discovery system output and with episodes (REFRACT findings built into HDD) |

*(Examples in brackets)

in Cognitive Science. The crucial point to realize is that the delimited phenomenon that is being studied is the behaviour of human scientists as they make discoveries (see Table 9.1, for direct comparison to Galileo). Hence, accounts of past and present episodes of scientific discovery, and the results of empirical psychological studies of human scientists, constitute the basic data of this domain. These data are considered as experimental tests results in terms of the framework. The account of the Galilean episode (Chapter 3) is a case in point. In the empirical studies scientific discovery domains have been simulated with varying degrees of realism (e.g. Wason's, 1960, '2 4 6' paradigm; Klahr & Dunbar's, 1988, computer controlled robot). These environments are experimental setups in the framework. They allow experimental tests to be carried out on the behaviour of scientists. The use of a computer simulated environments and specific pen-and-paper tasks are contrasting experimental paradigms (e.g. Reimann's, 1990, REFRACT program; *cf.* Qin & Simon's, 1990, equation finding task). There is communication between the experiments and theory; for example the findings of an empirical study may be used in the design of a computational model (e.g. Reimann, 1990, used his REFRACT findings to to build HDD) (see Table 9.1).

### 9.2.2 The Theoretical Component

In the theoretical component of his research on the motion of bodies, Galileo characterized the phenomena using hypotheses such as the Aristotelian effective weight laws and his own law of free fall. The models he inferred from these hypotheses applied to specific situations; for example the inclined plane. Both hypotheses and models were expressed as simple mathematical equations, so are examples of the state transformation conceptualization of theoretical knowledge. Galileo's predictions of motion down the inclined plane, generated by applying the model, are instances consisting of values for specified independent and dependent terms.

On the theoretical side of Cognitive Science, the abstract characterization of the

behaviour of the human scientist takes several forms. The theses proposed by philosophers of science, and the theories of some cognitive psychologists, are hypotheses. The computational models built in AI often instantiate such hypotheses and are considered as models. The specification of AI programs in terms of sets of heuristics and processes constitute state transformation functions. The transformation function's input state is given by the domain and inputs to the program on a particular run. The function's output is the description of how the program performs. Running a discovery system on a given domain generates an instance (see Table 9.1).

### 9.2.3 Criteria For The Acceptability Of Theories

Galileo assessed the acceptability of the free fall hypothesis in terms of the number of experimental paradigms to which it could be successfully applied. In AI the assessment of computer programs is typically in terms of performance (e.g. Kibler & Langley, 1988). However, when modelling scientific discovery in Cognitive Science the assessment of computational models is often considered in terms of the generality of the programs - explanatory breadth. The acceptability of computational models is something we will consider in more detail below.

We have seen how research in Cognitive Science into scientific discovery maps onto the framework. This neatly groups the various computer programs and different types of studies into distinct classes. This classification in turn reveals several different issues concerning the nature of the research on scientific discovery. We will consider: (§9.3) the limitations of the empirical research that has so far been carried out and how it may progress in the future; (§9.4) how computational models have, or very often have not, modelled the various components of scientific discovery; (§9.5) criteria for assessing the acceptability of discovery systems, and how well existing systems fared; (§9.6) other issues, for example, how the completeness of computational models may lead to emergent abilities.

## 9.3 THE EXPERIMENTAL COMPONENT OF THE COGNITIVE SCIENCE OF SCIENTIFIC DISCOVERY

According to the framework analysis, the experimental component of research in the study of scientific discovery in Cognitive Science is constituted by the historical episodes of discovery and the empirical research on human scientists. Both are involved in the development and assessment of hypotheses and computational models. How adequate they are for this purpose is an important issue in this field.

### 9.3.1 Historical Accounts Of Discovery

In terms of the framework accounts of episodes of discovery are mere observations; there is no manipulative control over the parameters that affect the discoveries made. This reduces substantially the reliability of the accounts as a means to assess or develop computational models or higher level characterizations. Without manipulative control of the phenomena, it is impossible to determine with certainty what has caused a scientist to perform a particular action.

It seems that the best that we can hope to achieve with historical cases studies is a detailed chronology of the events in an episode. The ordering of events is useful because a discovery system that manages to reproduce the same sequence of events is more likely to possess the right heuristics. In this sense we should prefer KEKEDA (Kulkarni & Simon, 1988) and STERN, over PI (Thagard, 1988), because they make discoveries in the same order as the episodes they model.

However, the approach is still susceptible to the vagaries of the historian's reconstruction of the course of events and even the model builder's interpretation of the published account. The empirical studies do not suffer from the same lack of manipulative control over the phenomena, but the are lacking in other ways.

### 9.3.2 Empirical Studies

Empirical research on scientific discovery has not had the same emphasis as the building of computational models. One explanation is that most investigators prefer not to get their hands "dirty" so have stuck to building computational models.

Another more technical explanation is that devising suitable simulated scientific discovery environments is difficult to do.

Many researchers have studied particular discovery tasks, these include: (i) how numerical data is generalized into laws (e.g. Gerwin, 1974); (ii) the solution of physics problems (e.g. Larkin *et.al.*, 1980); and (iii) the proposing of expressions to explain data (e.g. Wason, 1960). Although these studies have led to some interesting conclusions (such as the existence of confirmatory bias) it is by no means clear that the conclusions are applicable more generally beyond the narrow range of tasks considered.

One solution to such limitations is the construction of simulated scientific discovery environments (e.g. Klahr & Dunbar, 1988). The making of discoveries by subjects in these environments resembles the making of real discoveries more closely. To make a discovery the subjects have to carry out many different tasks, including the performing of experimental tests and the inference to and testing of laws. However, the simulated environments are not without their own limitations. The two following criticisms are partly related. First, the simulated environments over-constrain the behaviour of the subjects, distorting or preventing them from expressing the full range of processes they might normally display. Second, the environments only simulate one experimental setup. Thus the subjects can only make inference to models and not hypotheses. To be able to find general theories several experimental paradigms must be made available to the subject. Future work may address these deficiencies. The problem of the lack of experimental paradigms could be overcome in, for example, Klahr & Dunbar's (1988) computer controlled robot environment by giving the subject a number of mystery functions to characterize. The generalizing of all the descriptions of functions into a higher level characterizations would yield hypotheses.

The experimental component of scientific discovery research in Cognitive Science has been considered. We now more on to the theoretical component.

## 9.4 THE THEORETICAL COMPONENT OF THE COGNITIVE SCIENCE OF SCIENTIFIC DISCOVERY

The theoretical side of Cognitive Science research programmes on scientific discovery contains theoretical knowledge of several types. There are the theses of philosophers of science, some theories proposed by cognitive psychologists, and computational models in AI. Following the central theme of the thesis, our main concern will be with computational models of scientific discovery, with some attention to the conclusions drawn for empirical studies.

Amongst the computational modellers there have been some distinct biases. Many researchers have tended to be driven by specific AI techniques rather than being led by the episodes being modelled. The consequence of this is that important aspects of discovery have been ignored. In terms of the framework, discovery systems have typically focused on the models and instances of the episodes they consider. We will see how STERN has managed to overcome this and other limitations.

### 9.4.1 Theoretical Knowledge

AI has provided researchers with a means to model theoretical knowledge that no longer relies on the systems of logic used by philosophy of science (and all that that entails). The quality of the new representations is an important issue for the computational modelling of scientific discovery.

Early systems, like the programs in the BACON school, used simple representations of theoretical knowledge. Slightly later systems combined qualitative and quantitative representations into single programs (e.g. IDS, Nordhausen & Langley, 1987). Other systems have employed numerous classes of value-attribute pairs (e.g. KEKEDA, Kulkarni & Simon, 1988; HDD, Reimann, 1990). However, these representations have typically been domain-specific. Some

representations do not even consider the semantic contents of theories (e.g. Thagard, 1989a).

STERN is different. It makes a clear distinction between domain-specific and domain-independent knowledge. Domain-specific knowledge are formalisms such as the equations and qualforms used to model Galileo's knowledge. Domain-independent representations are the knowledge structures that instantiate the framework. Thus, in principle, it should only be necessary to change the domain-dependent formalisms (and rules) when modelling different domains (such as an episode from the history of chemistry).

To see whether this scheme in STERN works out in practice will require another domain to be modelled. There are many to choose from and herein lies another issue. The types of theoretical knowledge that have been modelled are ones that can be easily represented; such as arithmetic equations, chemical reactions, classes of objects and so forth. Non-trivial realistic representations of more complex, but also more interesting, types of theoretical knowledge have been avoided. A few examples are infinitesimal calculus, quantum mechanics, theories expressed propositionally (as in psychology), and even Galileo's geometric-pictorial representation. A full understanding of scientific discovery may not possible without the consideration of these more complex types of knowledge. For example, we may require specific processes to break up detailed hypotheses into manageable parts for testing.

### 9.4.2 Theoretical Inferences

A similar picture to that just considered with representations emerges with the modelling of theoretical inferences; this is not surprising as they are so closely related in discovery systems. Earlier systems tend to consider just one task (e.g. generalization from data to laws in the BACON school). Later programs combine quantitative and qualitative inference in particular tasks (e.g. ABACUS, Falkenhainer & Michalski, 1986). The most complete systems employ a number of

different processes that deal with one hypothesis or model at a time (e.g. SDDS, Klahr & Dunbar, 1988; HDD; KEKEDA). The hypotheses may be tested against experimental test results and modified according to the outcome.

STERN makes significant advances on previous systems, using three strategies to make theoretical inferences. The confirmation strategy not only tests quantitative and qualitative hypotheses against experimental tests but can make inferences with groups of mutually exclusive hypotheses (Chapter 6). The generalization strategy allows STERN to fully characterize the domain in a qualitative fashion, in which all relevant and irrelevant terms are identified. It also finds tentative quantitative hypotheses from the data. The strategy for the generation of new hypotheses from old considers qualitative and quantitative hypotheses using old hypotheses from the whole spectrum of acceptability.

Furthermore, STERN subsumes existing systems. The BACON programs have the same task as just one of STERN classes of rules; this is the generalizing of instances into models. The various types of knowledge and processes of HDD, SDDS and KEKEDA can also be mapped onto the components and rules of STERN.

Many different types of scientific reasoning have been modelled by STERN and other systems. However, there is one obvious omission – inferences using pictorial representations. Qin & Simon (1990) found that graphs are important in the generalization of data into laws, in particular to help identify the form of the function describing the data. Galileo used a geometric-pictorial method to generate models from hypotheses. Simon & Larkin (1987) have considered why graphical summaries of information are often so effective.

STERN roughly models some of these abilities. The function to assess predictive accuracy uses a least squares technique that is equivalent to the plotting of points and drawing a line though them (Chapter 5). STERN uses qualforms (i.e. regularities found in experimental data) to help choose the form of equations in the

strategy that generates new hypotheses from old. This is rather like the process of function spotting using diagrams that Qin & Simon observed. Clearly there is much interesting work yet to be done on this topic.

### 9.4.3 Acceptability Of Theories

How scientists ought to and how they actually assesses the acceptability of theories is something that philosophers have argued over long and hard. The builders of computational models have been more pragmatic and given their systems particular techniques to see if they are effective. The acceptability criterion typically considered by researchers are variations of explanatory breadth. STERN is not an exception in this respect and even ECHO (Thagard, 1989a) seems to boil down to it (see §2.4).

The various systems that assess the acceptability of theories have successfully modelled episodes of discovery using just the one criterion, but this is far form conclusive proof that it is a necessary and sufficient one. Intuitively, internal consistency and fruitfulness are two that seem to be relevant. Furthermore, McAllister (1989) contends that aesthetic criteria are important too. Watson (1968), one of the discoverer's of the structure of DNA, recollects that the beauty of the model substantially increased their belief in the double helix. Simplicity is one aesthetic criterion that has been considered at length by philosophers of science (e.g. Lakatos, 1971; Sober, 1975; McAllister, 1989) that has to some extent been taken on board in AI (e.g. Harman *et.al.*, 1988; Thagard, 1988a).

### 9.4.4 Experimental Knowledge

We now turn to experimental matters. The work on the theoretical component of scientific discovery outweighs the experimental component substantially. This parallels the neglect of experiment in the philosophy of science, that has begun to be rectified by the *new experimentalists* (e.g. Hacking, 1983; Galison, 1986; Franklin; 1987). In AI, STERN attempts to fulfil the same role, as we will see.

Almost without exception, the representation of experimental knowledge in discovery systems in absent. Typically the programs take interpreted, true, noise-free data as input. In the exceptional cases only the experimental test level of the experimental component of the framework is considered (i.e. KEKEDA, HDD).

STERN on the other hand considers experimental paradigms, experimental setups and experimental tests. Many types of experimental paradigms and setups are given to STERN as inputs. STERN also considers new paradigms and setups by itself. All three levels of experiments have associated information; such as pragmatic measures of their relative ease of manufacture and use. Different types of experiments are also acknowledged; *normal* experiments (e.g. pendulum paradigm) and *combined* experiments (e.g. inclined plane and projectile paradigm) used in one of two *modes*. Like terms in theory, the experimental components have parameters. Parameters are not simply variables with assigned values but include information on (i) the range over which its values may vary and (ii) how easy the parameters are to manipulate and observe. Furthermore, STERN distinguishes between experiments that it has only conceived and those that have been manufactured.

### 9.4.5 Experimental Processes

Since most previous systems do not represent experiments they do not instantiate any experimental processes. HDD represents experiments, but all its experimental tests are given as inputs. KEKEDA designs domain-specific experimental tests but only when there is some instance to be tested.

The experimental processes in STERN are extensive. Unlike KEKEDA, it can design experiments when there is no instance to be tested (and when there is). When there is no instance the wealth of experimental knowledge is used to generate all possible tests and the background knowledge is used to eliminate designs that are trivial. STERN uses various rules for the selection of experimental paradigms and setups. The selection depends on (i) the hypothesis (or model) being tested, if any,

(ii) the number of setups possessed by the paradigm, and (iii) whether the hypothesis and paradigm have been considered together earlier. Furthermore, STERN limits the manufactured experimental paradigms to a manageable number. Typically previous systems do not distinguish between experimental tests and instances. STERN, however, can compare experimental tests and predictive instances, and is able to interpret tests into instances. The pinnacle of STERN's experimental abilities is the construction of new experiments by combining known experimental paradigms. STERN does this when there are theories that are intractable from a wholly theoretical approach. This is a good example of the interaction between theory and experiment in STERN.

### 9.4.6 Reliability Of Experiments

The framework for scientific discovery acknowledges the importance of the assessment of the reliability of experimental knowledge. This is something that the new experimentalists in the philosophy of science have shown to be crucial in scientific discovery. One aspect of experimental reliability concerns the presence of noise in experimental data and how to deal with it.

BACON uses a simple mechanism to cope with noise, but it is rather crude (see §2.2.1.1). Other systems have assumed that experimental data is noise free.

STERN deals with noise in different ways depending on the strategy being followed. During the confirmation strategy STERN compares predictive instances with experimental results. The function that assesses predictive accuracy takes noise into account, the greater the noise the lower the degree of accuracy. In the strategy that generalizes experimental results into hypotheses noise is considered as instances that are generalized to form models. A model is only formed when a deviation test, that ranges over all the data, is satisfied.

The many other strategies discovered by the new experimentalists for dealing with experimental reliability have been analysed by Cheng (1988) using the black box conceptualization of experiments.

### 9.4.7 Communication

Communication between the theoretical and experimental components of a research programme is essential to scientific discovery. The implicit view of most researchers is that the only type of communication is the feeding of experimental data into theoretical considerations. However, in research programmes there is information transfer between all the levels of the two main components for all sorts of different reasons. On a mundane level, to test a prediction an experimenter needs to know which parameters are relevant when designing and performing an experimental test. More interestingly, there must have been some subtle interchanges between theory and experiments for Galileo to know that invention of combined inclined plane and projectile paradigm would allow the, otherwise intractable, law of free fall to be tested.

In previous work experiments are almost completely ignored, so there has been little modelling of communication. However, KEKEDA and HDD are both able to compare predictions and experimental results.

Since the theory and experiment components are instantiated in STERN many types of communication occur between the levels of theory and experiment. The confirmation of a hypothesis requires continual exchange of information (Chapter 5) and the generalization strategy requires a moderate amount of communication (Chapter 6). The correspondence between theoretical terms and experimental parameters is assumed to be the most fundamental level of communication by the framework. STERN models this in its rich representation of experimental parameters and theoretical terms.

### 9.4.8 Background Knowledge

Background knowledge certainly has a role in scientific discovery - the scientific life of researchers is not isolated from the rest of their knowledge and experiences. Galileo is a good case in point; he used his knowledge of geometry in many

different ways to make discoveries.

Sleeman *et.al.*'s (1989) architecture for theory driven discovery acknowledges the importance of background knowledge. However, the architecture does not spell out how such knowledge is to be distinguished from the other forms of knowledge considered (e.g. meta-knowledge), nor does it indicate in detail how the knowledge is used. Holland *et.al.*'s Induction framework states that background knowledge is important. However, the PI program instantiation of the Induction framework does not seem to use such knowledge (Thagard, 1988a). None of the previous systems actually employ background knowledge to make discoveries.

In STERN background knowledge can be defined as any knowledge that does not fall within the theoretical or experimental components. Such knowledge may still be scientific, in a general sense, but it will be background information as far as the delimited set phenomena of the domain is concerned. STERN possesses background knowledge of geometry and relations for spherical bodies. This knowledge is used in the generation of models to form hypotheses and in the design of experiments.

We have considered a long list of features that we would expect an acceptable model of scientific discovery to instantiate. The extent to which models cover these features is one way to assess the acceptability of the models. We will now consider this and other criteria for the assessment of computational models.

## 9.5 THE ACCEPTABILITY OF COMPUTATIONAL MODELS OF SCIENTIFIC DISCOVERY

We have considered the investigation of scientific discovery in Cognitive Science as a scientific pursuit in terms of the framework. One of the aspects of the theoretical component of the framework are criteria by which to assess the acceptability of theories. In this section we will consider three important criteria.

### 9.5.1 Completeness

In the previous section we considered many different aspects of the framework that computational models of scientific discovery ought to consider if they are to successfully model all parts of complex episodes of discovery in detail. The degree to which a model does this can be considered as a measure of its *completeness*. This way of assessing the acceptability of discovery systems does not seem to have been considered before; perhaps because no one has produced a model as complete as STERN.

Completeness is an important property for a discovery system to possess. Without it, it is impossible for a system to model an episode of discovery in any detail. Furthermore, completeness allows the modeller to consider tasks and heuristics at a higher level than before - at the level of the main tasks and processes within the research programme being modelled. There are interesting types of behaviour to be studied at this level that are not manifested in less complete and hence less complex systems (see §9.6.1).

Along the completeness dimension of acceptability the previous discovery systems are clustered at the incomplete end. For example, programs in the BACON school only generalize instances into models. ECHO only assesses the acceptability of propositions in terms of data. Although KEKEDA has many classes of heuristics it does not consider hypotheses, experimental setups and paradigms. STERN, however, is fairly complete as it instantiates all the components of the framework, that is: the three levels of theory and experiments; theoretical inferences and experimental processes; criteria for the acceptability of theories and the reliability of experiment; and communication between theories and experiments.

### 9.5.2 Generality

Another important criterion is *generality*; that is the range of different domains to which a model can be validly applied - explanatory breadth. This is a criterion that has been acknowledge in this field and which some researchers have striven to

achieve (e.g. Thagard, 1988). Like any theoretical knowledge, the more domains a computational model validly applies to the more likely it is to be a characterization that captures some underlying essence of scientific discovery phenomena.

BACON and ECHO are two systems that have generality, both have been applied successfully to many domains. However, this generality is at the expense of being very incomplete. Other programs have much less generality. For instance, GELL-MANN (Zytkow, 1987) only considers the inference to quark models from data, and KEKEDA only models the discovery of the urea cycle; although in KEKEDA's favour it is moderately complete.

STERN has only modelled the Galilean domain so far, so it has not been shown to be general. However, various facts indicate that it has potential to satisfy this acceptability criterion. First, STERN is based heavily on the framework, that is in turn quite general. Second, STERN's knowledge structures and processes largely subsume HDD, SDDS and KEKEDA that each relate to quite different domains. Third, STERN's knowledge structures and classes of rules have been carefully separated into domain-specific and domain-independent groups. This allows the domain-independent rules to be tested using other domains just by swapping the domain-specific formalisms and rules (at least in principle). The real test of STERN's generality will be to try to model other episodes of discovery in the future.

### 9.5.3 Internal Coherence

Applying the framework in a reflexive fashion, to the Cognitive Science of scientific discovery, focuses our attention on the relation between the various theoretical elements. These elements are the computational models and any general hypotheses about scientific discovery. Three examples are: (i) the BACON school's cyclic general description of science and the programs in the school (Langley *et.al.*, 1987); (ii) the Theory of Explanatory Coherence and ECHO (Thagard, 1989a); and

closer to home, (iii) the framework for scientific discovery and STERN.

In this respect, there is another important criterion for assessing the acceptability of computational systems. This time it is the internal relationship between the computational model and the hypothesis which is of interest, rather than external reference to discovery episodes. In a Cognitive Science research programme that includes a computational model, the *internal coherence* of the theoretical component (of the research programme) is to a large extent determined by the quality of the relationship between its hypothesis and its computational model. The computational model should follow fully and rationally from the hypothesis; in other words the relationship should: (i) not be arbitrary or involve *ad hoc* assumptions; and (ii) have all the essential features of the hypothesis instantiated in the model. If these conditions are not satisfied, then the acceptability of the model cannot be used to assess the generality of the hypothesis.

Let us compare the BACON school with the research in this thesis. They are areas of research with very different degrees of internal coherence.

The general cyclic description of the BACON school considers four phases including: (i) data gathering; (ii) finding parsimonious descriptions of the data; (iii) formulating explanatory theories; and (iv) testing these theories. Of the four phases outlined, only the second (formation of parsimonious descriptions of data) phase is actually modelled in the programs. Furthermore, separate programs are required to deal with different domains. The school is far from internally coherent.

The framework in this thesis has been implemented in STERN. Everything posited by the framework reappears in the program (See Chapters 1 & 4 and §9.4). This is not only in terms of knowledge structures but includes inference processes and acceptability and reliability criteria. The work here has internal coherence. Furthermore, this coherence is clearly demonstrated by the reflexive abilities of the framework - STERN is based on it within the present research programme and the research programme is itself characterized by the framework.

This completes the consideration of issues arising out the analysis of Cognitive Science research on scientific discovery using the framework. We will now consider other issues, some of which relate indirectly to what we have just considered.

## 9.6 REMAINING ISSUES AND THOUGHTS

We have considered various issues to do with computational models and how successful they have been so far at modelling scientific discovery. Now we will consider some tentative conclusions drawn from the performance of STERN and the best of the previous systems.

### 9.6.1 Completeness Leads To Emergent & High Level Abilities

One of the criterion for the assessment of the acceptability of computational models considered above was *completeness*. Apart from the fact that it gives more exact representations of scientific research programmes, there are two other related reasons why completeness is important.

First, a complete model possesses many different processes that instantiate particular tasks that make up the overall ability of a system. Compared with less complete models that only model one task, the more complete system allows processes and heuristics amongst the tasks to be considered. These higher or research programme level procedures are particularly interesting. Included amongst them are approaches that scientists use to speed up their research. For example, scientists often prefer "dirty" but fast approaches as in Crick and Watson's discovery of the Double Helix structure of DNA. They used the quickest, if not the most reliable, methods because they were attempting to beat Pauling to a discovery worthy of a Nobel Prize. Platt (1964) has formalized some heuristics derived from this episode into a prescriptive method he calls "strong inference".

In the most complete discovery systems such research programme heuristics have begun to be studied. KEKEDA has measures of the amount of effort spent on

certain tasks. They are used to help control the path of discovery by forcing the system to abandon propositions when there has been too much work for too little reward, even if there is insufficient evidence to show the proposition is really unacceptable. KEKEDA can also recognize surprising outcomes and places a task to study the surprising effect at the top of its agenda; thus abandoning what it was previous investigating. STERN possess two heuristics at the level of the research programme to cut down the amount of search. First, the program groups together mutually exclusive hypotheses that were generated by a single process. Later, when one hypothesis is confirmed, STERN knows that the rest can simply be ignored. Second, the mechanism to control the availability of the experiments eliminates much wasted effort during the confirmation and the generalization strategies.

The second reason why completeness in a computational model is desirable is related to the first. Completeness allows the programmer to build in high level procedures, but it also means that discovery systems will start to exhibit behaviour that is emergent as they become more and more complex. By emergent behaviour we mean that the performance of the systems is no longer simple to predict from its structure and inputs. For example, STERN found the law governing the period of swing of a pendulum, even though there had been no intention on the part of STERN's designer that it would find this true law. Another, perhaps more significant, behaviour that is exhibited in STERN is *chronological dependency.*

STERN has four main strategies that use many processes to perform many different tasks to gradually increase and modify the knowledge about the phenomena. These processes are dependent on what previous processes have done. When a process modifies the knowledge in the system this indirectly affects later processes that use the knowledge. For example, in STERN the initial acceptability of the Aristotelian instantaneous acceleration law permitted the term for speed to be eliminated from equations using the term's definition. A bit later the instant

acceleration law was disconfirmed. Thus, when STERN came to consider the free fall equation it found it could not replace the speed term. Finally, to overcome the intractability of the theoretical inferences STERN considered the combined inclined plane and projectile experiment. It would have been difficult to predict that this long line of many different processes would have occurred just given the specification of STERN's four main strategies and the Aristotelian laws as input. This clearly shows how interesting behaviour over time emerges from complex and complete systems.

The completeness of models is clearly an important aspect to be pursued in the modelling of discovery. But there are also many other aspects of scientific discovery that are interesting to study.

### 9.6.2 Beyond The Single Scientist And Research Programme

The framework for scientific discovery has been at the heart of this thesis. So far we have considered the framework as applied to a single scientist working more or less in isolation with the theoretical and experimental components fixed in several ways. A great deal has already been achieved using the framework in this manner, but loosening up some of the present constraints will allow the framework's potential to be explored even more fully.

The framework may be applied to research programmes that involve cooperating or competing scientists investigating a common class of phenomena. Such domains could be modelled by giving STERN one complete scientific knowledge hierarchy to represent the knowledge possessed by each scientist (see Figure 1.2). Interactions between the two would then be modelled using heuristics like those found by Frankel (1987) in his analysis of the continental drift debate. Frankel's heuristics consider how to challenge the work of opponents whilst improving one's own position.

The framework may also be used to model the reasons scientists have for working in a particular domain and why they choose to abandon or switch to

another domains. For example, Crick (1988) decided to work in molecular biology by applying his "gossip test". This test was based on which fields of research were being talked about the most by scientists in general. Galileo switched research programmes when he learnt of the invention of the telescope. Perhaps astronomy fascinated him more, but it is also reasonable to consider whether discoveries in the natural motion domain were drying up. STERN might use some mechanism to measure the rate at which hypotheses and models were being generated to assess the fruitfulness of a domain.

These are just two examples of how the framework may be applied more generally than has so far been considered in this thesis.

## 9.7 CONCLUSION

A great deal has been achieved in this thesis. A framework for characterizing scientific discovery has been introduced. It has been used to organize a review of previous work in this area of Cognitive Science, and to some extent assess the acceptability of existing computational models. The STERN discovery system was based on the framework and has successfully modelled in detail the discoveries of an important historical episode of science. STERN overcomes the limitations of existing discovery systems, particularly with respect to the criteria of completeness and internal coherence.

# References

Buchanan, B.G. & Feigenbaum, E.A. (1978). Dendral and MetaDendral: Their applications dimension. *Artificial Intelligence*, **11**, 5-24.

Bundy, A., Byrd., L., Luger., G., Mellish., C. & Palmer, M. (1979). Solving mechanics problems using meta-level inference. In Michie, D. (ed). *Expert Systems in the Micro-electronic Age*. Edinburgh: Edinburgh University Press.

Campbell, A.N., Hollister, V.F., Duda, R.O., Hart, P.E. (1982). Recognition of a hidden mineral deposit by an artificial intelligence program. *Science*, **217**(3),927-9.

Cheng, P.C-H. (1988). The neglect of experiment. Master's degree dissertation. Department of Philosophy, University of Warwick.

Cheng, P.C-H. (1989a). Bridging the divide: A framework for scientific discovery. Paper presented at the *Evolving Knowledge in Natural Science and Artificial Intelligence* Conference, Reading, September, 1989.

Cheng, P.C-H (1989b). The genesis of a computational scientific theorist and experimental researcher. Technical Report No.57, Human Cognition Research Laboratory, The Open University, Milton Keynes, England.

Cheng, P.C-H (1990). A Scientific Research Programme In Artificial Intelligence For Computational Modelling Of Scientific Discovery. Human Cognition Research Laboratory, The Open University, Milton Keynes, England.

Cheng, P.C-H. & Keane, M.T. (1989a). A framework for scientific discovery. *Cognitive systems*, **24**.

Cheng, P.C-H. & Keane, M.T. (1989b), Explaining explanatory coherence psychologically. *Behavioral and Brain Science*, **12**(3), 470-1

Cohn, A.G. (1989). Approaches to qualitative reasoning. *Artificial Intelligence Review*, **3**,177-232.

Crick, F. (1988). *What a Mad Pursuit: A Personal View of Scientific Discovery.* New York: Basic Books.

Davis, R. & Lenat, D.B. (1982). *Knowledge-based systems in Artificial Intelligence*, New York: McGraw-Hill.

Drake, S. (1973a). Galileo's discovery of the law of free fall. *Scientific American*, **228**,85-92.

Drake, S. (1973b). Galileo's experimental confirmation of horizontal inertia: unpublished manuscripts (Galileo gleanings XXII). *ISIS*, **64**(233),291-305.

Drake, S. (1975).Galileo's new science of motion. In Bonelli, M.L.R., & Shea, W.R. *Reason, Experiment, and Mysticism.* London: Macmillan Press.

Drake, S. & MacLachlan, J. (1975). Galileo's Discovery of the parabolic trajectory. *Scientific American.* **232**(3),102-110.

Duda, R., Gashing, J., & Hart, P. (1979). Model design in the Prospector consultant system for mineral exploration. In Michie, D. *Expert systems in the Micro-electronic Age.* Edinburgh: Edinburgh University Press.

Evans, J. (1989). *Bias in human reasoning: causes and consequences.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Falkenhainer, B.C. & Michalski, R.S. (1986). Integrating quantitative and qualitative discovery: the ABACUS system. *Machine Learning*, **1**(4),367-401.

Fisher, P. & Zytkow, J.M. (forthcoming). Discovering quarks and other hidden objects. Paper submitted to the Machine Learning Conference, 1990.

Feigl, H. (1970). The 'Orthodox' view of theories: remarks in defense as well as critique. In Radner, M. & Winokur, S. *Minnesota Studies in the Philosophy of Science.* vol.4. Minneapolis: University of Minnesota Press.

Feyerabend, P.K. (1975). *Against Method.* London: New Left Books.

Feynman, R. (1965). *The Character of Physical Law*. London: BBC books.

Frankel, H. (1987). The continental drift debate. In Engelhardt, H.T.jr. & Caplan, A.L. (eds), *Scientific Controversies*. Cambridge: Cambridge University Press.

Franklin, A. (1986). *The Neglect of Experiment*. Cambridge: Cambridge University Press.

Friedland, P. & Kedes, L.H. (1985). Discovering the secrets of DNA. *Communications of the ACM*, **28**(11),1164-86.

Galileo (1838, 1954) *Dialogues Concerning two new sciences*. New York:Dover.

Galison, P. (1987). *How Experiments End*. Chicago: The University of Chicago Press.

Gerwin, D. (1974). Information processing, data inference, and scientific generalization. *Behavioral Science*, **19**, 314-25.

Giere, R.N. (1989). *Explaining Science: a cognitive approach*. Chicago:University of Chicago press.

Gooding, D., Pinch, T., & Schaffer, S. (eds), (1989). *The uses of experiment; Studies in the natural sciences*. Cambridge: Cambridge University Press

Gorman, M.E. (1989). Error, falsification, and scientific inference; an experimental investigation. *Quarterly journal of experimental psychology*, **41A** (2), *385-412*.

Gray, N.A.B. (1984). Applications of artificial intelligence for organic chemistry:analysis of C-13 spectra. *Artificial Intelligence*, **22**, 1-21.

Gray, N.A.B., Sleeman, D.H., Stacey, M.K. (1988). Machine discovery and the operationalization of scientific theories. Technical report AUCS/TR8801, Department of Computing Science, University of Aberdeen.

Hacking, I. (1983). *Representing and Intervening*. Cambridge: Cambridge University Press.

Harman, G., Ranney, M., Salem, K., Döring, F., Epstein, J., Jarorska, A. (1988). A theory of simplicity. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*. Hillsdale NJ: Lawrence Erlbaum.

Hayes-Roth, B., Buchanan, B., Lichtarge, O., Hewett, M., Altman, R., Brinkley, J., *et.al.*(1986). PROTEAN: deriving protein structure from constraints, in *Proceedings of the fifth national conference of artificial intelligence.*

Hasemer, T. & Domingue, J.(1989). *Common LISP programming for artificial intelligence*. Wokingham, England:Addison-Wesley.

Hill, D.K. (1988). Galileo's early experiments on projectile motion and the law of free fall. *ISIS*, **79**,646-68.

Holten, G.& Roller, D. (1958). *Foundations of Modern Physical Science*. Read Mass: Addison-Wesley.

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

Humphreys, W.C. (1967). Galileo, falling bodies and inclined planes. *British Journal for the History of Science*, **3**(11),225-44.

Jones, R. & Langley, P. (1988), A theory of scientific problem solving, in *Proceedings of the tenth annual conference of cognitive science society*, Montreal Canada, August 1988. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Kibler, D. & Langley, P. (1988). Machine learning as experimental science. In Sleeman, D. (ed). *Proceedings of the third European Working Session on Learning*. London: Pitman.

Klahr, D. & Dunbar, K. (1988) Dual space search during scientific reasoning. *Cognitive Science*, **12**, 1-48.

Koehn, B.W. & Zytkow, J.M. (1986). Experimenting and theorizing in theory formation. In Raz, Z. & Zemankova, M., (eds). *Proceeding of the international symposium on methodologies for intelligent systems*, Knoxville TN:ACM Sigart Press, 296-307.

Kokar, M.M. (1986). Determining arguments of invariant functional descriptions. *Machine Learning*, **1**(4),403-422.

Koyre, A. (1968). *Metaphysics and measurement.* London:Chapman & Hall.

Kuhn, T.S. (1970). *The Structure of Scientific Revolutions.* 2nd ed. Chicago: University of Chicago Press.

Kulkarni, D. & Simon, H.A. (1988). The processes of scientific discovery: the strategy of experimentation. *Cognitive Science*, **12**, 139-75.

Lakatos, I. (1971). History of science and its rational reconstructions. In Buck R.C. & Cohen, R.S. (eds). *Boston Studies in the Philosophy of Science, Vol 8.* Dordrecht: Reidel.

Lakatos, I. (1974). Falsification and the methodology of scientific research programmes. In Lakatos, I. & Musgrave, A., (eds). *Criticism and the Growth of Knowledge.* Cambridge: Cambridge University Press.

Langley, P., Bradshaw, G.L., & Simon, H.A. (1984). Rediscovering chemistry with the BACON system. In Michalski R.S., Carbonell, J.G., & Mitchell, T.M., (eds) *Machine Learning An Artificial Intelligence Approach.*New York: Springer-Verlag.

Langley, P., Zytkow, J.M., Simon, H.A., & Bradshaw, G.L. (1986). The search for regularity: four aspects of scientific discovery. In Michalski R.S., Carbonell, J.G., & Mitchell, T.M., (eds). *Machine Learning An Artificial Intelligence Approach II.* Los Altos, California: Morgan Kaufmann.New York: Springer-Verlag.

Langley, P., Simon, H.A., Bradshaw, G.L. & Zytkow, J.M., (1987) *Scientific Discovery Computational Explorations of the Creative Process.* Cambridge, Mass.: MIT Press.

Langley, P. & Zytkow, J.M. (1989) Data-driven approaches to empirical discovery. *Artificial Intelligence*, **40**, 283-312.

Larkin, J.H., McDermott, J., Simon, D.P. & Simon H.A. (1980). Models of competence in solving physics problems. *Cognitive Science*, **4**,317-345.

Lenat, D.B. (1983). EURISKO: A program that learns new heuristics and domain concepts. *Artificial Intelligence*, **21**, 68-98.

Lenat, D.B. & Brown, J.S. (1984). Why AM and EURISKO appear to work. *Artificial Intelligence*, **23**, 269-94.

Luger, G.F. (1981). Mathematical model building in the solution of mechanics problems: human protocols and the Mecho Trace. *Cognitive Science*, **5**,55-77.

MacLachlan, J. (1976). Galileo's experiments with pendulums: real and imaginary. *Annals of Science*. **33**,173-85.

Magie, W.F. (1935). *A Source book in Physics*. Cambridge, MA: Harvard University Press.

McAllister, J.W. (1989). Truth and beauty in scientific reason. *Synthese*, **78**,25-51.

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

Mitchell, T.M., Keller, R.M. & Kedar-Cabelli, S.T. (1986). Explanation-based generalization: a unifying view. *Machine learning*, **1**,47-80.

Naylor, R.H. (1974). Galileo and the problem of free fall. *The British Journal For The History Of Science*, **7** (26), 105-34.

Naylor, R.H. (1975). An aspect of Galileo's study of the parabolic trajectory. *ISIS*, **66**,394-6.

Naylor, R.H. (1976). Galileo: the search for the parabolic trajectory. *Annals of Science*, **33**,153-72.

Nordhausen, B. & Langley, P.(1987). Towards an integrated discovery system. *Proceedings of the Tenth. International Joint Conference on Artificial Intelligence.*

Platt, J.R. (1964), Strong Inference. *Science* **146**(3642),347-53.

Popper, K. (1959). *The Logic Of Scientific Discovery*. London: Hutchinson.

Popper, K. (1965). *Conjectures And Refutations: The Growth Of Scientific Knowledge*. 2nd ed. New York: Basic Books.

Qin, Y. & Simon, H.A. (1990). Laboratory replication of scientific discovery processes. *Cognitive Science*, **14**, 281-312.

Rajamoney, S., Dejong, G., & Faltings, B. (1985). Towards a model of conceptual knowledge acquisition through directed experimentation. In *Proceeding of the 9th International joint Conference on Artificial Intelligence.* Los Altos, CA: Morgan Kaufmann. 688-690.

Ranney, M. & Thagard, P. (1988). Explanatory coherence and belief revision in naive physics. In *Proceedings of the Tenth. Annual Conference of the Cognitive Science Society*. Montreal, Canada.

Reimann, P. (1990). *Problem Solving Models of Scientific Discovery Learning Processes*. Frankfurt am Main: Peter Lang.

Rose, D. (1988a). Discovery and Revision via Incremental Hill Climbing. *Proceedings of the International Workshop on Machine Learning and Meta Reasoning.*

Rose, D. (1988b). Multiple Theories In Scientific Discovery. In *Proceedings Tenth Annual Conference Of The Cognitive Science Society*. Hillsdale, New Jersey: Lawrence Erlbaum. 695-701.

Rose, D. & Langley, P. (1986). Chemical discovery as belief revision. *Machine Learning*, **1**(4),423-52.

Settle, T.B. (1961). An experiment in the history of science. *Science*, **133** (19-23).

Simon, H.A. & Larkin, J.H. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, **11**, 65-99. And in Simon, H.A. (1989). *Models of Thought*. Vol II. New Haven: Yale University Press.

Sleeman, D.H., Stacey, M.K., Edwards, P., & Gray, N.A.B. (1989). An architecture for theory-driven scientific discovery. In Morik, K. (ed). *Proceedings of the 4th EWSL*. London: Pitman. 11-23.

Sober, E. (1975). *Simplicity*. Oxford:Clarendon Press.

Suppe, F. (1977). *The Structure of Scientific Theories*. 2nd.ed. Urbana: University of Illinois Press.

Thagard, P. (1988a). *Computational Philosophy of Science*. Cambridge, Massachusetts: Massachusetts Institute of Technology Press.

Thagard, P. (1988b), Explanatory coherence. CSL Report No.16. Cognitive Science Laboratory, Princeton University.

Thagard, P. (1989a). Explanatory coherence. *Behavioral and Brain Sciences*, **12** (3), 435-67.

Thagard, P. (1989b), The dinosaur debate: explanatory coherence and the problem of competing hypotheses. CSL Report. Cognitive Science Laboratory, Princeton University.

Thagard, P. (in press). The conceptual structure of the chemical revolution. *Philosophy of Science*. June 1990.

Thagard, P. (forthcoming). Conceptual Revolutions. Manuscript under review.

Thagard, P. & Holyoak, K.J. (1985). Discovering the wave theory of sound. In *Proceedings of the Ninth. International Joint Conference on Artificial Intelligence*. Los Altos, California: Kaufmann.

Walker, M.G. (1987). How feasible is automated discovery? *IEEE Expert*, **2**(1), 69-82.

Wason, P.C. (1960) On the failure to eliminate hypotheses in a conceptual task. *Quarterly journal of experimental psychology*, **12**, 129-40.

Watson, J.D. (1968). *The Double Helix: a personal account of the discovery of the structure of DNA*. Harmondsworth, Middlesex: Penguin

Zytkow, J.M. (1987) Combining many searches in the FAHRENHEIT discovery system. In *Proceedings Fourth International Workshop on Machine Learning*. Irvine CA, 281-87.

Zytkow, J.M. & Simon, H.A. (1988). Normative systems of discovery and logic of search. *Synthese*, **74**, 65-90.

# Appendix I

## STERN's Discovery Trace

The output trace produced by STERN when modelling the Galilean episode is presented below in a condensed form.

The cycles of the Production System (PS) begin on the numbered lines. The first figure is the cycle number and the second in square brackets is the depth PS nesting. On each cycle the matched rules are named in the parentheses and one chosen by conflict resolution is shown after the arrow (=>). The statements following the numbered lines give a brief description of the actions performed; the amount of indentation of these lines provides an additional indication the level of PS nesting.

The inputs to STERN are briefly discussed in §4.2.1.

### I.1 Disconfirming The Aristotelian Laws

STERN is run by an initial call to the PS with RULES_0. In the first cycle the R0_START_CONFIRM rule is chosen from amongst the alternatives that have successfully matched. The action of this rule sends STERN down the route of hypothesis testing through the generation of models by a recursive call to the PS with RULES_1.

```
   +++  P.S. called with RULES_0   +++
1-[0] (R0_NEW_EXPTPARADIGMS R0_START_CONFIRM) => R0_START_CONFIRM
   Calling production system with rules_1
     +++  P.S. called with RULES_1   +++
```

One of the three Aristotelian hypotheses is chosen in cycle 2. The pendulum experimental paradigm is selected on cycle 3. A recursive call to the PS with RULES_5, cycle 4, permits the generation of models, in cycle 5. The model is inferred by substituting the definition of speed for the velocity term and the quotient of weight and volume for the density term. No further rules match at in the present PS level, [2], so control is returned to the PS with RULES_1 on level [1], in cycle

6. The final action of the rule begun on cycle 4 is to store the new model.

```
2-[1] (R1_CHOOSE_PARADIGM_NO_HYPO R1_CHOOSE_HYPO) =>
      R1_CHOOSE_HYPO
     hypo with (= T_V T_DEN) made current
3-[1] (R1_HYPO_ASSESS R1_CHOOSE_PARADIGM_WITH_HYPO) =>
      R1_CHOOSE_PARADIGM_WITH_HYPO
     PENDULUM made current
4-[1] (R1_HYPO_ASSESS R1_HYPO->MODELS) => R1_HYPO->MODELS
     Calling production system with RULES_5 to generate models
       +++  P.S. called with RULES_5   +++
5-[2] (R5_MAIN_WORKER) => R5_MAIN_WORKER
       model with equation (= (/ T_D T_TIME) (/ T_W T_VOL))
       added to hypo's list and tractability now
       #S(MEASURE NUMBER 1 DEGREE 1)
6-[2] NIL => no rule to fire
       --- No rules in RULES_5 to fire ---
     1 model(s) generated and stored
```

STERN now makes the model active, cycle 7, and proceeds to investigate it by calling RULES_2, cycle 8. The first action at the new level is the choice of an experimental setup from amongst those stored in the current experimental paradigm, cycle 9. Instances are then generated by RULES_8 in a further recursive call to the PS. Six instances are found, cycle 11, and control is popped back up a level. The instances are stored and the measure of tractability of the model is incremented, cycle 12.

```
7-[1] (R1_HYPO->MODELS R1_CHOOSE_MODEL) => R1_CHOOSE_MODEL
     model with equation (= (/ T_D T_TIME) (/ T_W T_VOL)) made
     current
8-[1] (R1_TEST_MODEL) => R1_TEST_MODEL
     call production system with RULES_2
       +++  P.S. called with RULES_2   +++
9-[2] (R2_CHOOSE_SETUP) => R2_CHOOSE_SETUP
       DOWN_PENDULUM made current
10-[2] (R2_OBTAIN_INSTANCES) => R2_OBTAIN_INSTANCES
       Calling production system with RULES_8 to generate
       instances
         +++  P.S. called with RULES_8   +++
11-[3] (R8_GEN_QUANT_INSTANCES) => R8_GEN_QUANT_INSTANCES
         6 instance(s) and stored in current instance
12-[3] NIL => no rule to fire
         --- No rules in RULES_8 to fire ---
       model tractability now #S(MEASURE NUMBER 1 DEGREE 1)
       6 instance(s) made and stored in model, current instance
       cleared
```

Each instances is chosen in turn for comparison with an experimental test in cycles 13-20, 21-28, 29-36, 37-47, 48-58, and 56-66. The first and fourth sets of cycles will be considered as they are examples of the two patterns of instance testing. First cycles 13-20, that shows the failure to test an instance. At cycle 13, distance and

time are chosen as independent and dependent terms, and the PS is called with RULES_9 to consider instances and experimental tests, cycle 14. Yet another recursive call is made to invoke the experimental performance rule, cycles 15 and 16. However, no experiment is performed, cycle 17, because it is difficult to measure distances for swinging pendulums. So the adequacy of the instance is set to -1 to indicate the fact that no experimental results were available, cycle 18. No further experimental performance rules can fire, cycle 19, so the instance is simply made inactive and model adequacy is not incremented, cycle 20.

```
13-[2] (R2_CHOOSE_INSTANCE) => R2_CHOOSE_INSTANCE
       An instance with these variables made current
       independent = T_D and dependent = T_TIME
       their values are respectively -
       ( 0.000  0.337  0.673  1.010  1.346  1.683  2.020  2.356 )
       ( 0.000  0.000  0.001  0.001  0.001  0.002  0.002  0.002 )
14-[2] (R2_TEST_INSTANCE) => R2_TEST_INSTANCE
       call production system with rules_9 to test the instance
         +++  P.S. called with RULES_9   +++
15-[3] (R9_PERFORM_EXPT_TEST) => R9_PERFORM_EXPT_TEST
         Calling production system to perform experiment
           +++  P.S. called with EXPT_RULES   +++
16-[4] (E_PREPARE_WITH_INSTANCE) => E_PREPARE_WITH_INSTANCE
           Test prepared
17-[4] NIL => no rule to fire
           --- No rules in EXPT_RULES to fire ---
         Test Performed
18-[3] (R9_NO_TEST_PERFORMED) => R9_NO_TEST_PERFORMED
         Instance degree set to -1 and current expt test cleared
19-[3] NIL => no rule to fire
         --- No rules in RULES_9 to fire ---
       degree of match of instance and expt. test -1
20-[2] (R2_ASSESS_MODEL) => R2_ASSESS_MODEL
       Model tractable but current instance not testable.
       Current instance cleared.
   . . . . . .
```

The second set of cycles, 37-47, is the first to successfully compare an instance with experimental test results. The instance with time and weight terms is chosen, cycle 37, and the PS is called twice in succession, cycles 38 and 39, to test the instance that in turn requires the performance of an experiment. The down_pendulum experiment is performed and produces a list of experimental weight and time parameter values, cycles 40 to 42. The comparison of the two pairs of values is the task of RULES_12, cycles 43 to 45, which finds there is no correlation (weight and time being independent). Cycle 46 ends the instance testing, and the adequacy of the model is calculated, cycle 47.

```
37-[2] (R2_CHOOSE_INSTANCE) => R2_CHOOSE_INSTANCE
       An instance with these variables made current
       independent = T_W and dependent = T_TIME
       their values are respectively -
       ( 0.000  0.143  0.286  0.429  0.571  0.714  0.857  1.000 )
       (58904862254808624.000  0.004  0.002  0.001  0.001  0.001
        0.001  0.001 )
38-[2] (R2_TEST_INSTANCE) => R2_TEST_INSTANCE
       call production system with rules_9 to test the instance
         +++  P.S. called with RULES_9   +++
39-[3] (R9_PERFORM_EXPT_TEST) => R9_PERFORM_EXPT_TEST
         Calling production system to perform experiment
           +++  P.S. called with EXPT_RULES   +++
40-[4] (E_PREPARE_WITH_INSTANCE) => E_PREPARE_WITH_INSTANCE
           Test prepared
41-[4] (E_DOWN_PENDULUM) => E_DOWN_PENDULUM
           TIME values found for o/p values of current expttest
42-[4] NIL => no rule to fire
           --- No rules in EXPT_RULES to fire ---
         Test Performed
         i/p* and o/p vals are -
         ( 0.000  0.143  0.286  0.429  0.571  0.714  0.857  1.000)
         ( 0.710  0.687  0.696  0.707  0.687  0.697  0.690  0.693)
43-[3] (R9_TEST_INSTANCE) => R9_TEST_INSTANCE
         Calling production system with RULES_12
         to compared instance and expttest
           +++  P.S. called with RULES_12   +++
44-[4] (R12_TEST_QUANT_INSTANCE) => R12_TEST_QUANT_INSTANCE
           degree of match =  0.000
45-[4] NIL => no rule to fire
           --- No rules in RULES_12 to fire ---
         Degree of agreement between instance and expttest
         =  0.000
         Current expt test cleared
46-[3] NIL => no rule to fire
         --- No rules in RULES_9 to fire ---
       degree of match of instance and expt. test 0
47-[2] (R2_ASSESS_MODEL) => R2_ASSESS_MODEL
       Model adequacy now #S(MEASURE NUMBER 1 DEGREE 0)
       & current instance cleared.
  . . . . . .
```

Once RULES_2 has cycled through all the instances, the active experimental setup is dropped, cycle 67. Beginning with the selection of the SWING_PENDULUM setup, cycles 68 to 126, the whole process of testing a model against an experimental setup is repeated (as in cycles 9 to 67). Control is passed back up to RULES_1 in cycle 127, and the adequacy of the model is given; four instances have been tested but none compared well with experimental results. No more suitable setups exist under the active paradigm so the adequacy of the hypothesis is incremented, cycle 128.

```
67-[2] (R2_NO_MORE_INSTANCES) => R2_NO_MORE_INSTANCES
       current setup cleared
68-[2] (R2_CHOOSE_SETUP) => R2_CHOOSE_SETUP
       SWING_PENDULUM made current
  . . . . . .
```

```
126-[2] (R2_NO_MORE_INSTANCES) => R2_NO_MORE_INSTANCES
       current setup cleared
127-[2] NIL => no rule to fire
       --- No rules in RULES_2 to fire ---
     model adequacy now #S(MEASURE NUMBER 4 DEGREE 0)
128-[1] (R1_HYPO_ASSESS_WRT_MODELS) => R1_HYPO_ASSESS_WRT_MODELS
     hypo adequacy = #S(MEASURE NUMBER 1 DEGREE 0) and current
     model cleared.
     All models tested, exptparadigm cleared
```

The whole hypothesis testing process is repeated with the INCPLANE experimental paradigm (as per cycles 3 to 128) in cycles 129 to 197. The new model has the same form of equation as that in cycle 7. The comparison of the instance, with distance and time as independent and dependent terms, with its experimental test was good. The degree slot of the adequacy measure of the model is significantly above zero.

```
129-[1] (R1_HYPO_ASSESS R1_CHOOSE_PARADIGM_WITH_HYPO) =>
        R1_CHOOSE_PARADIGM_WITH_HYPO
     INCPLANE made current
130-[1] (R1_HYPO_ASSESS R1_HYPO->MODELS) => R1_HYPO->MODELS
     Calling production system with RULES_5 to generate models
  . . . . . .
     model adequacy now
     #S(MEASURE NUMBER 3 DEGREE 0.96635554951792)
  . . . . . .
```

The hypothesis has thus been tested against two experimental paradigms and their various setups. The adequacy of the hypothesis is finally calculated in cycle 198. This is below the adequacy limit, so it is abandoned. The next Aristotelian hypothesis is chosen, cycle 200, and it is tested in a similar manner (a repeat of cycles 2 to 199) using the same two experimental paradigms to generate models, cycles 200 to 386.

```
198-[1] (R1_HYPO_ASSESS_WRT_MODELS) => R1_HYPO_ASSESS_WRT_MODELS
     hypo adequacy =
     #S(MEASURE NUMBER 2 DEGREE 0.3221185165059733)
     and current model cleared
     All models tested, exptparadigm cleared
199-[1] (R1_HYPO_ASSESS) => R1_HYPO_ASSESS
     current hypo cleared
200-[1] (R1_CHOOSE_PARADIGM_NO_HYPO R1_CHOOSE_HYPO) =>
R1_CHOOSE_HYPO
     hypo with (= T_V T_W*) made current
  . . . . . .
385-[1] (R1_HYPO_ASSESS_WRT_MODELS) => R1_HYPO_ASSESS_WRT_MODELS
     hypo adequacy = #S(MEASURE NUMBER 2 DEGREE 0) and current
     model cleared
     All models tested, exptparadigm cleared
```

```
386-[1] (R1_HYPO_ASSESS) => R1_HYPO_ASSESS
     current hypo cleared
```

The third Aristotelian law is the qualitative instantaneous acceleration hypothesis. The procedure has a similar structure to the two previous quantitative hypotheses except that qualitative heuristics for the domain specific generative processes are employed. The testing of the hypothesis with the first experimental setup of the first paradigm is shown below, cycles 387 to 409. The cycles 410 to 425 cover the second experimental setup and cycles 428 to 450 cover the inclined plane experimental paradigm.

```
387-[1] (R1_CHOOSE_PARADIGM_NO_HYPO R1_CHOOSE_HYPO) =>
        R1_CHOOSE_HYPO
     hypo with (INSTANTANEOUS T_V T_D) made current
388-[1] (R1_HYPO_ASSESS R1_CHOOSE_PARADIGM_WITH_HYPO) =>
        R1_CHOOSE_PARADIGM_WITH_HYPO
     PENDULUM made current
389-[1] (R1_HYPO_ASSESS R1_HYPO->MODELS) => R1_HYPO->MODELS
     Calling production system with RULES_5 to generate models
       +++  P.S. called with RULES_5   +++
390-[2] (R5_COMPLEX_QUALFORM) => R5_COMPLEX_QUALFORM
       model made with (INSTANTANEOUS T_V T_D) as qualform
391-[2] NIL => no rule to fire
       --- No rules in RULES_5 to fire ---
     1 model(s) generated and stored
392-[1] (R1_HYPO->MODELS R1_CHOOSE_MODEL) => R1_CHOOSE_MODEL
     model with qualform (INSTANTANEOUS T_V T_D) made current
393-[1] (R1_TEST_MODEL) => R1_TEST_MODEL
     call production system with RULES_2
       +++  P.S. called with RULES_2   +++
394-[2] (R2_CHOOSE_SETUP) => R2_CHOOSE_SETUP
       DOWN_PENDULUM made current
395-[2] (R2_OBTAIN_INSTANCES) => R2_OBTAIN_INSTANCES
       Calling production system with RULES_8 to generate
       instances
         +++  P.S. called with RULES_8   +++
396-[3] (R8_GEN_QUAL_INSTANCES) => R8_GEN_QUAL_INSTANCES
         1 qualitative instance made and stored in current
         instance as a list
397-[3] NIL => no rule to fire
         --- No rules in RULES_8 to fire ---
       model tractability now #S(MEASURE NUMBER 1 DEGREE 1)
       1 instance(s) made and stored in model, current instance
       cleared
398-[2] (R2_CHOOSE_INSTANCE) => R2_CHOOSE_INSTANCE
       An instance with these variables made current
       independent = T_D and dependent = T_V
399-[2] (R2_TEST_INSTANCE) => R2_TEST_INSTANCE
       call production system with rules_9 to test the instance
         +++  P.S. called with RULES_9   +++
400-[3] (R9_PERFORM_EXPT_TEST) => R9_PERFORM_EXPT_TEST
         Calling production system to perform experiment
           +++  P.S. called with EXPT_RULES   +++
401-[4] (E_PREPARE_WITH_INSTANCE) => E_PREPARE_WITH_INSTANCE
```

```
          Test prepared
402-[4] (E_DOWN_PENDULUM) => E_DOWN_PENDULUM
          SPEED values found for o/p values of current expttest
403-[4] NIL => no rule to fire
          --- No rules in EXPT_RULES to fire ---
         Test Performed
404-[3] (R9_TEST_INSTANCE) => R9_TEST_INSTANCE
         Calling production system with RULES_12
         to compared instance and expttest
           +++  P.S. called with RULES_12   +++
405-[4] (R12_TEST_QUAL_INSTANCE) => R12_TEST_QUAL_INSTANCE
          Qualform not matched
406-[4] NIL => no rule to fire
          --- No rules in RULES_12 to fire ---
         Degree of agreement between instance and expttest
         =  0.000
         Current expt test cleared
407-[3] NIL => no rule to fire
         --- No rules in RULES_9 to fire ---
       degree of match of instance and expt. test 0
408-[2] (R2_ASSESS_MODEL) => R2_ASSESS_MODEL
       Model adequacy now #S(MEASURE NUMBER 1 DEGREE 0)
       & current instance cleared.
409-[2] (R2_NO_MORE_INSTANCES) => R2_NO_MORE_INSTANCES
       current setup cleared
  . . . . . .
     model adequacy now #S(MEASURE NUMBER 2 DEGREE 0)
427-[1] (R1_HYPO_ASSESS_WRT_MODELS) => R1_HYPO_ASSESS_WRT_MODELS
     hypo adequacy = #S(MEASURE NUMBER 2 DEGREE 1) and current
model cleared
     All models tested, exptparadigm cleared
  . . . . . .
```

The final adequacy of the qualitative Aristotelian hypothesis is calculated at cycle 451.

```
451-[1] (R1_HYPO_ASSESS_WRT_MODELS) => R1_HYPO_ASSESS_WRT_MODELS
     hypo adequacy = #S(MEASURE NUMBER 3 DEGREE 1) and current
     model cleared.
     All models tested, exptparadigm cleared
```

The whole discomfirmation procedure is finished by cycle 452. All the Aristotelian hypotheses have been experimentally tested and found to be unacceptable. So the RULES_1 has nothing else to do but return control to RULES_0, which now decides to follow the generalization strategy.

```
452-[1] (R1_HYPO_ASSESS) => R1_HYPO_ASSESS
     current hypo cleared
453-[1] (R1_CHOOSE_PARADIGM_NO_HYPO) => R1_CHOOSE_PARADIGM_NO_HYPO
     PENDULUM paradigm made current
454-[1] NIL => no rule to fire
     --- No rules in RULES_1 to fire ---
   Finished testing existing hypos
```

## I.2 Generalizing Experiments Into Hypotheses

When generalizing experimental findings into hypotheses it is necessary to first obtain models and instances. To begin an experimental paradigm is chosen, cycle 456, and the PS is called with RULES_7 in order to obtain models, cycle 457. RULES_7 prepares an model frame, with unfilled slots, and selects the DOWN_PENDULUM experimental setup is chosen, cycle 458. Experimental tests are prepared for all the possible permutations of experimental parameters, cycle 459, and then whittled down to a set of reasonable combinations, cycle 460.

```
455-[0] (R0_NEW_HYPOS R0_START_INDUCE) => R0_START_INDUCE
   current paradigm cleared
   Calling production system with RULES_3
     +++  P.S. called with RULES_3   +++
456-[1] (R3_CHOOSE_PARADIGM) => R3_CHOOSE_PARADIGM
     New current hypo made
     Selected paradigm PENDULUM
457-[1] (R3_GET_MODELS) => R3_GET_MODELS
     Calling production system with rules_7 to get models
       +++  P.S. called with RULES_7   +++
458-[2] (R7_CHOOSE_SETUP) => R7_CHOOSE_SETUP
       A current model made
       expt. setup DOWN_PENDULUM chosen
459-[2] (R7_OBTAIN_EXPTTESTS) => R7_OBTAIN_EXPTTESTS
       27 expt. tests made
460-[2] (R7_EXPTTEST_PREFERENCES) => R7_EXPTTEST_PREFERENCES
       15 expt. tests stored in setup and current expt test
       cleared
```

The performance and analysis of the fifteen different experimental setups falls into two patterns, as exemplified by the two sets of cycles, 461 to 465, and 466 to 476. In the first set no suitable experimental tests are performed. The test is chosen, cycle 461, and the PS is called with RULES_11 to handle instances, cycle 462. This in turn calls the PS is with the experimental rules, but no test can be performed, cycle 463 and 464, so control is return to RULES_7.

```
461-[2] (R7_CHOOSE_EXPTTEST) => R7_CHOOSE_EXPTTEST
       Expt test with i/p* as DISTANCE and o/p as TIME removed
       from expt setup and made current
462-[2] (R7_MAKE_INSTANCES) => R7_MAKE_INSTANCES
       Calling production system with rules_11 to make instance(s)
         +++  P.S. called with RULES_11   +++
463-[3] (R11_PERFORM_EXPT_TEST) => R11_PERFORM_EXPT_TEST
         Calling production system to perform experiment
           +++  P.S. called with EXPT_RULES   +++
464-[4] NIL => no rule to fire
           --- No rules in EXPT_RULES to fire ---
         current exptest cleared
465-[3] NIL => no rule to fire
         --- No rules in RULES_11 to fire ---
       0 instance(s) made and made current.
```

```
Current expt. test cleared.
```

In the second pattern of cycles, STERN successfully performs an experimental test and interprets the results to obtain instances which are stored. The experimental test has height and time as input and output parameters, respectively, cycle 466. Again the PS is called successively with RULES_7 and EXPT_RULES to obtain instances and experimental tests, cycles 467 and 468. The appropriate experimental test is performed, cycles 469 and 470. RULES_6 is invoked to to interpreted the test results into instances, cycles 471 to 473. The new instances are stored, cycles 475-6.

```
466-[2] (R7_CHOOSE_EXPTTEST) => R7_CHOOSE_EXPTTEST
       Expt test with i/p* as HEIGHT and o/p as TIME removed from
       expt setup and made current
467-[2] (R7_MAKE_INSTANCES) => R7_MAKE_INSTANCES
       Calling production system with rules_11 to make instance(s)
         +++  P.S. called with RULES_11   +++
468-[3] (R11_PERFORM_EXPT_TEST) => R11_PERFORM_EXPT_TEST
         Calling production system to perform experiment
           +++  P.S. called with EXPT_RULES   +++
469-[4] (E_DOWN_PENDULUM) => E_DOWN_PENDULUM
           TIME values found for o/p values of current expttest
470-[4] NIL => no rule to fire
           --- No rules in EXPT_RULES to fire ---
         i/p* and o/p vals are -
         ( 0.000  0.179  0.357  0.536  0.714  0.893  1.071  1.250)
         ( 0.000  0.234  0.334  0.404  0.493  0.562  0.654  0.758)
471-[3] (R11_INTERP_TO_INST) => R11_INTERP_TO_INST
         Calling production system with RULES_6
         to expttest and make instance(s)
           +++  P.S. called with RULES_6   +++
472-[4] (R6_FIND_QUALFORMS) => R6_FIND_QUALFORMS
           Instance with qualform (INCREASE T_TIME T_H)
           made and appended to current instance
           Instance with qualform (FROM_ZERO T_TIME T_H)
           made and appended to current instance
473-[4] (R6_SIMPLE_TRANSFER) => R6_SIMPLE_TRANSFER
           Instance made with values copied from expt test and
           appended to current instance.
           Current expt test cleared.
474-[4] NIL => no rule to fire
           --- No rules in RULES_6 to fire ---
         3 instance(s) made
475-[3] NIL => no rule to fire
         --- No rules in RULES_11 to fire ---
       3 instance(s) made and made current.
       Current expt. test cleared.
476-[2] (R7_CHOOSE_EXPTTEST R7_STORE_INSTANCES) =>
        R7_STORE_INSTANCES
       3 instance(s) now stored in current model
       current instance cleared
  . . . . . .
```

The rest of the experimental tests are examined one of the two ways considered. Twenty two instances are found. The whole processes, from cycle 458 to 599, is then repeated with the next experimental setup of the current paradigm, starting at cycle 600 and ending at 742.

```
600-[2] (R7_GENERALISE_INSTANCES R7_CHOOSE_SETUP) =>
        R7_CHOOSE_SETUP
       expt. setup SWING_PENDULUM chosen
601-[2] (R7_GENERALISE_INSTANCES R7_OBTAIN_EXPTTESTS) =>
        R7_OBTAIN_EXPTTESTS
       27 expt. tests made
602-[2] (R7_GENERALISE_INSTANCES R7_EXPTTEST_PREFERENCES) =>
        R7_EXPTTEST_PREFERENCES
       15 expt. tests stored in setup and current
       expt test cleared
  . . . . . .
       51 instance(s) now stored in current model
       current instance cleared
742-[2] (R7_GENERALISE_INSTANCES) => R7_GENERALISE_INSTANCES
       Calling production system with RULES_4 to generalise
       instances
```

STERN now calls the PS with RULES_4 to generalise the 51 instances. Thirty three models are found, 32 with qualforms and 1 with an equation, cycles 743 to 747. The models are stored, cycle 748. The quantitative model is in fact the correct law describing the relation between the length of a pendulum and its period of swing.

```
        +++  P.S. called with RULES_4   +++
743-[3] (R4_PREPARE) => R4_PREPARE
        51 instance(s) moved from model and made current.
        Current model instances cleared.
744-[3] (R4_MODEL_EQNS R4_MODEL_QUAL) => R4_MODEL_QUAL
        Instances with qualforms removed from current
        instance and generalised to models with qualforms
        (FROM_ZERO T_V T_@) (REPEAT+ T_V T_@) (REPEAT+ T_V T_S)
        (REPEAT+ T_V T_VOL) (STEADY T_V T_VOL) (REPEAT- T_V T_W)
        (STEADY T_V T_W) (FROM_ZERO T_V T_L) (REPEAT+ T_V T_L)
        (FROM_ZERO T_V T_H) (REPEAT+ T_V T_H)
        (FROM_ZERO T_TIME T_@) (REPEAT+ T_TIME T_@)
        (REPEAT+ T_TIME T_S) (REPEAT- T_TIME T_VOL)
        (STEADY T_TIME T_VOL) (REPEAT+ T_TIME T_W)
        (STEADY T_TIME T_W) (FROM_ZERO T_TIME T_L)
        (REPEAT+ T_TIME T_L) (FROM_ZERO T_TIME T_H)
        (REPEAT+ T_TIME T_H) (INCREASE T_V T_@)
        (INCREASE T_V T_S) (REPEAT- T_V T_VOL) (INCREASE T_V T_W)
        (INCREASE T_V T_L) (INCREASE T_V T_H)
        (INCREASE T_TIME T_@) (INCREASE T_TIME T_S)
        (INCREASE T_TIME T_L) (INCREASE T_TIME T_H)
745-[3] (R4_MODEL_EQNS) => R4_MODEL_EQNS
        1 model(s) made with equations (= T_S (* T_TIME T_TIME))
        Current instance cleared
746-[3] NIL => no rule to fire
```

```
        --- No rules in RULES_4 to fire ---
      33 model(s) made and current expt paradigm cleared
747-[2] NIL => no rule to fire
      --- No rules in RULES_7 to fire ---
    33 model(s) made
748-[1] (R3_STORE_MODELS) => R3_STORE_MODELS
    Models store in current hypo and current model
    and expt paradigm cleared
  . . . . . .
```

The inclined plane experimental paradigm is now chosen and the whole generalization processes to obtain models is repeated (as in cycles 456 to 748). Beginning at cycle 749, producing and storing 18 qualitative model and 3 quantitative models, by cycle 878.

```
874-[3] (R4_MODEL_EQNS R4_MODEL_QUAL) => R4_MODEL_QUAL
        Instances with qualforms removed from current
        instance and generalised to models with qualforms
        (STEADY T_V T_VOL) (INCREASE T_V T_VOL) (STEADY T_V T_W)
        (FROM_ZERO T_V T_L) (INCREASE T_V T_L)
        (FROM_ZERO T_V T_H) (INCREASE T_V T_H)
        (FROM_ZERO T_V T_D) (INCREASE T_V T_D)
        (STEADY T_TIME T_VOL) (STEADY T_TIME T_W)
        (INCREASE T_TIME T_W) (FROM_ZERO T_TIME T_L)
        (INCREASE T_TIME T_L) (FROM_ZERO T_TIME T_H)
        (INCREASE T_TIME T_H) (FROM_ZERO T_TIME T_D)
        (INCREASE T_TIME T_D)
875-[3] (R4_MODEL_EQNS) => R4_MODEL_EQNS
        3 model(s) made with equations
        (= T_D (* T_TIME T_TIME)) (= T_H (* T_TIME T_TIME))
        (= T_L (* T_TIME T_TIME))
        Current instance cleared
876-[3] NIL => no rule to fire
        --- No rules in RULES_4 to fire ---
      21 model(s) made and current expt paradigm cleared
  . . . . . .
```

The models just stored and the ones obtained for the pendulum experimental paradigm are now generalised into hypotheses, cycles 879 to 884, using RULES_10; 12 qualitative and 4 quantitative ones are made.

```
879-[1] (R3_GENERALISE_MODELS) => R3_GENERALISE_MODELS
    Calling production system with RULES_10 to generalise models
      +++  P.S. called with RULES_10   +++
880-[2] (R10_PREPARE) => R10_PREPARE
      54 model(s) copied from hypo and made current.
      Current hypo models cleared.
881-[2] (R10_HYPO_EQNS R10_HYPO_QUALS) => R10_HYPO_QUALS
      Models with qualforms removed from current model and
      generalised to hypos with qualforms
      (INCREASE T_TIME T_H) (FROM_ZERO T_TIME T_H)
      (INCREASE T_TIME T_L) (FROM_ZERO T_TIME T_L)
      (STEADY T_TIME T_W) (STEADY T_TIME T_VOL)
      (INCREASE T_V T_H) (FROM_ZERO T_V T_H) (INCREASE T_V T_L)
```

```
       (FROM_ZERO T_V T_L) (STEADY T_V T_W) (STEADY T_V T_VOL)
882-[2] (R10_HYPO_EQNS) => R10_HYPO_EQNS
       4 hypo(s) made with equations
       (= T_S (* T_TIME T_TIME)) (= T_D (* T_TIME T_TIME))
       (= T_H (* T_TIME T_TIME)) (= T_L (* T_TIME T_TIME))
       Current model cleared
883-[2] NIL => no rule to fire
       --- No rules in RULES_10 to fire ---
     16 hypo(s) made. Current hypo and exptparadigm cleared
884-[1] NIL => no rule to fire
     --- No rules in RULES_3 to fire ---
   Finished trying generalise new hypos strategy
   16 hypo(s) made and stored. Current hypo cleared
```

These hypotheses are adequate, because they have been successfully inferred from experimental results. They are also now candidates for the confirmation generative strategy, as considered in §I.1 above. However, only the quantitative hypotheses were not generalised from both manufacture experimental paradigms, so only they are tested. In cycles 885 to 890 STERN attempts to test the first new quantitative hypothesis against the inclined plane experiment. However, it fails to generate a model because the size term (T_S) has no equivalent experimental parameter.

```
885-[0] (R0_NEW_HYPOS R0_START_CONFIRM) => R0_START_CONFIRM
   Calling production system with rules_1
     +++  P.S. called with RULES_1   +++
886-[1] (R1_CHOOSE_PARADIGM_NO_HYPO R1_CHOOSE_HYPO) =>
        R1_CHOOSE_HYPO
     hypo with (= T_S (* T_TIME T_TIME)) made current
887-[1] (R1_HYPO_ASSESS R1_CHOOSE_PARADIGM_WITH_HYPO) =>
        R1_CHOOSE_PARADIGM_WITH_HYPO
     INCPLANE made current
888-[1] (R1_HYPO_ASSESS R1_HYPO->MODELS) => R1_HYPO->MODELS
     Calling production system with RULES_5 to generate models
       +++  P.S. called with RULES_5   +++
889-[2] (R5_MAIN_WORKER) => R5_MAIN_WORKER
       no model equation made, tractability now
       #S(MEASURE NUMBER 2 DEGREE 1)
       and current hypo cleared
890-[2] NIL => no rule to fire
       --- No rules in RULES_5 to fire ---
     current paradigm cleared
  . . . . . .
```

The other three quantitative hypotheses are tested in turn in cycles 885-890, 891-926, and 927-999. Fewer instances and experimental tests are need because there a many fewer terms in the equations of the hypotheses. When finished there are no more hypotheses that need testing so control is passed back up to the top most level, cycle 1000.

```
1000-[1] NIL => no rule to fire
     --- No rules in RULES_1 to fire ---
   Finished testing existing hypos
```

## I.3 New Hypotheses From Old

Both the generative and the generalization strategies have run their course with all the stored hypotheses and experimental paradigms, so STERN attempts to infer new hypotheses from the existing ones by invoking RULES_14, cycle 1001. Fourteen quantitative hypotheses are made, cycle 1002. Control is returned to RULES_0, cycles 1003.

```
1001-[0] (R0_NEW_HYPOS) => R0_NEW_HYPOS
   Calling production system with RULES_14
   to make new hypos form existing ones
     +++  P.S. called with RULES_14   +++
1002-[1] (R14_QUAL_TO_EQNS) => R14_QUAL_TO_EQNS
     14 hypo(s) made and stored, with equations
     (= T_V (EXPT T_H 1/2)) (= T_V (EXPT T_H 1/3))
     (= T_V (EXPT T_H 2)) (= T_V (EXPT T_H 2/3))
     (= T_V (EXPT T_H 3)) (= T_V (EXPT T_H 3/2)) (= T_V T_H)
     (= T_V (EXPT T_L 1/2)) (= T_V (EXPT T_L 1/3))
     (= T_V (EXPT T_L 2)) (= T_V (EXPT T_L 2/3))
     (= T_V (EXPT T_L 3)) (= T_V (EXPT T_L 3/2)) (= T_V T_L)
1003-[1] NIL => no rule to fire
     --- No rules in RULES_14 to fire ---
   Finished trying generate new hypos
   and current expt paradigm cleared, if any
```

Untested hypotheses now exist, so STERN tries to confirm all fourteen in turn. Cycles 1004 to 1009 are the set of cycles for the first hypothesis with the pendulum experimental paradigm; it is representative of actions performed on the same hypothesis with the inclined plane paradigm, and other the other thirteen hypotheses, up to cycle 1146. From the chosen hypothesis, cycle 1005, and an experimental paradigm, cycle 1006, an attempt is attempt is made to generate models, cycle 1007. However, this fails because the speed term can no longer be replaced by its definitional equation (as in eg. cycle 5), as none of its necessary qualitative conditions apply any longer (ie. the Aristotelian instantaneous acceleration law was disconfirmed earlier). Thus the hypothesis tractability is incremented the experimental paradigm cleared, cycles 1008 and 1009.

```
1004-[0] (R0_NEW_EXPTPARADIGMS R0_START_CONFIRM) =>
   R0_START_CONFIRM
   Calling production system with rules_1
     +++  P.S. called with RULES_1   +++
1005-[1] (R1_CHOOSE_PARADIGM_NO_HYPO R1_CHOOSE_HYPO) =>
     R1_CHOOSE_HYPO
     hypo with (= T_V (EXPT T_H 1/2)) made current
```

```
1006-[1] (R1_HYPO_ASSESS R1_CHOOSE_PARADIGM_WITH_HYPO) =>
     R1_CHOOSE_PARADIGM_WITH_HYPO
     PENDULUM made current
1007-[1] (R1_HYPO_ASSESS R1_HYPO->MODELS) => R1_HYPO->MODELS
     Calling production system with RULES_5 to generate models
       +++  P.S. called with RULES_5   +++
1008-[2] (R5_MAIN_WORKER) => R5_MAIN_WORKER
       no model equation made, tractability now
       #S(MEASURE NUMBER 1 DEGREE 0)
       and current hypo cleared
1009-[2] NIL => no rule to fire
       --- No rules in RULES_5 to fire ---
     current paradigm cleared
  . . . . . .
1146-[1] NIL => no rule to fire
     --- No rules in RULES_1 to fire ---
   Finished testing existing hypos
```

## I.4 New Experimental Paradigms

All the hypotheses have now been tested with all the stored experimental paradigms. So STERN decides to consider new experimental paradigms by combining existing paradigms using RULES_13, cycle 1147. This process involves: choosing a terminal, cycle 1148; finding suitable initial parts, cycle 1149; and making experimental setups, cycle 1150. The process is repeated five more times, in cycles 1152 to 1176, with other experimental paradigms as terminals.

```
1147-[0] (R0_NEW_EXPTPARADIGMS) => R0_NEW_EXPTPARADIGMS
   Calling production system with RULES_13 to make new
   expt paradigms
     +++  P.S. called with RULES_13   +++
1148-[1] (R13_REDUCE_*_THRESHOLD R13_CHOOSE_TERMINAL) =>
     R13_CHOOSE_TERMINAL
     PENDULUM chosen as current expt. paradigm
1149-[1] (R13_MAKE_COMBINES) => R13_MAKE_COMBINES
     PENDULUM+PENDULUM  combined expt paradigms created
1150-[1] (R13_COMBINED_SETUPS) => R13_COMBINED_SETUPS
     Expt setups made for each expt paradigm
1151-[1] (R13_COMBINED_SETUPS) => no rule to fire
     --- No rules in RULES_13 to fire ---
   1 expt paradigms made and stored
  . . . . . .
```

Of the six newly invented experimental paradigms, and the simple ones that have not previously been considered, one must be selected for manufacture and use. In cycles 1177 to 1179 RULES_13, reduces the value of the pragmatic parameter of manufacture ease and number of setups to a value that just brings one new experimental paradigm into reach. The paradigm is the combined inclined plane and projectile experiment.

```
1177-[0] (R0_NEW_EXPTPARADIGMS) => R0_NEW_EXPTPARADIGMS
   Calling production system with RULES_13 to make new expt
paradigms
     +++  P.S. called with RULES_13   +++
1178-[1] (R13_REDUCE_*_THRESHOLD) => R13_REDUCE_*_THRESHOLD
     Expt manf. ease and setup product reduced to 0.6000
1179-[1] (R13_REDUCE_*_THRESHOLD) => no rule to fire
     --- No rules in RULES_13 to fire ---
   0 expt paradigms made and stored
```

The confirmation strategy takes over once again, cycle 1180, starting with the size time hypothesis originating from the pendulum experiments. Cycles 1181 to 1185 are a repeat of cycles 886 to 890, in which no model is generated, and the tractability is simply amended.

```
1180-[0] (R0_NEW_EXPTPARADIGMS R0_START_INDUCE
         R0_START_CONFIRM) => R0_START_CONFIRM
  . . . . . . . . . . . . . . . . . . . .
```

The next hypothesis to be chosen is in fact the free fall hypothesis, cycle 1186. The combined experiment is chosen, cycle 1187, and two models generated, cycles 1188 to 1190. One model applies to each of the modes of the combined experimental paradigm.

```
1186-[1] (R1_CHOOSE_PARADIGM_NO_HYPO R1_CHOOSE_HYPO) =>
     R1_CHOOSE_HYPO
     hypo with (= T_V (EXPT T_H 1/2)) made current
1187-[1] (R1_HYPO_ASSESS R1_CHOOSE_PARADIGM_WITH_HYPO) =>
         R1_CHOOSE_PARADIGM_WITH_HYPO
     INCPLANE+PROJECTILE made current
1188-[1] (R1_HYPO_ASSESS R1_HYPO->MODELS) => R1_HYPO->MODELS
     Calling production system with RULES_5 to generate models
       +++  P.S. called with RULES_5   +++
1189-[2] (R5_COMB_COMPLEX_EQN) => R5_COMB_COMPLEX_EQN
       Model of INITIAL type, with
       (= (EXPT T_H 1/2) (/ T_L T_TIME)) as equation.
       Model of TERMINAL type, with
       (= (EXPT T_H 1/2) (/ T_L T_TIME)) as equation
1190-[2] NIL => no rule to fire
       --- No rules in RULES_5 to fire ---
     2 model(s) generated and stored
```

The initial mode model is chosen for examination first, cycle 1191. In cycles 1192 to 1260, the model is tested in the established manner. For the combined model it is possible to eliminate the quantitative theoretical term for speed, because the speed at the end of the ramp is equal to the horizontal speed of the projectile. The comparison of the instance and experimental test is good so the acceptability of the model is set accordingly, cycle 1260.

```
1191-[1] (R1_HYPO->MODELS R1_CHOOSE_MODEL) => R1_CHOOSE_MODEL
     model with equation (= (EXPT T_H 1/2) (/ T_L T_TIME)) made
current
     Model is of INITIAL type
1192-[1] (R1_TEST_MODEL) => R1_TEST_MODEL
     call production system with RULES_2
  . . . . . .
1260-[1] (R1_HYPO_ASSESS_WRT_MODELS) => R1_HYPO_ASSESS_WRT_MODELS
     hypo adequacy =
     #S(MEASURE NUMBER 1 DEGREE 0.9299284370149901)
     and current model cleared
```

The terminal model is chosen and tested in the same way as the first, cycles 1261 to 1329. The input and output experimental parameters in this experiment are both from the terminal part of the experimental setup. As parameters are the height and length this model describes the parabolic flight path of projectiles.

```
1261-[1] (R1_CHOOSE_MODEL) => R1_CHOOSE_MODEL
     model with equation (= (EXPT T_H 1/2) (/ T_L T_TIME)) made
     current
     Model is of TERMINAL type
1262-[1] (R1_TEST_MODEL) => R1_TEST_MODEL
     call production system with RULES_2
  . . . . . .
1329-[2] NIL => no rule to fire
       --- No rules in RULES_2 to fire ---
     model adequacy now
     #S(MEASURE NUMBER 1 DEGREE 0.9320678881749285)
```

The final acceptability of the hypothesis is high as both models were themselves acceptable.

```
1330-[1] (R1_HYPO_ASSESS_WRT_MODELS) => R1_HYPO_ASSESS_WRT_MODELS
     hypo adequacy = #S(MEASURE NUMBER 2 DEGREE 1.861996325189919)
     and current model cleared
     All models tested, exptparadigm cleared
```

The current hypothesis is acceptable so its associated hypotheses (with `T_H` & `T_TIME` terms) are unacceptable. The heuristic concerning grouped hypotheses generated by the same procedure thus applies and sets the measure of acceptability of the associated hypotheses appropriately, cycle 1331. This saves considerable effort as the six related hypotheses are not examined. It is purely an accident of ordering of the hypotheses (by RULES_14) that the Galilean hypothesis was investigated first. Had it been stored further down the list other hypotheses wound have been considered first.

```
1331-[1] (R1_HYPO_ASSESS) => R1_HYPO_ASSESS
     Adequacy of hypotheses related to current hypo
```

```
     decremented because current hypo adequate.
     current hypo cleared
```

STERN now tests all the hypotheses with the speed and length terms, starting at cycle 1332 and ending at cycle 1589. Again two models are generated for each, but none of the hypotheses are found to be adequate. STERN has to plough though every hypothesis, because there is no effort saving heuristic applicable for these disconfirmed hypotheses.

```
1332-[1] (R1_CHOOSE_PARADIGM_NO_HYPO R1_CHOOSE_HYPO) =>
R1_CHOOSE_HYPO
     hypo with (= T_V (EXPT T_L 1/2)) made current
  . . . . . . . . . . . . . . . . . . .
```

STERN has now modelled the main elements of the Galilean episode. Galileo stopped at this point because his attention was diverted by the invention of the telescope. However, STERN continues by attempting to confirm the qualitative hypotheses that have not been tried with the combined experimental paradigm, cycles 1598 onwards. The experiment performance rules have almost been exhausted, so STERN will simply find that it cannot obtain experimental results.

```
1598-[1] (R1_CHOOSE_PARADIGM_NO_HYPO R1_CHOOSE_HYPO) =>
     R1_CHOOSE_HYPO
     hypo with (INCREASE T_TIME T_H) made current
1599-[1] (R1_HYPO_ASSESS R1_CHOOSE_PARADIGM_WITH_HYPO) =>
     R1_CHOOSE_PARADIGM_WITH_HYPO
     INCPLANE+PROJECTILE made current
1600-[1] (R1_HYPO_ASSESS R1_HYPO->MODELS) => R1_HYPO->MODELS
     Calling production system with RULES_5 to generate models
  . . . . . . . . . . . . . . . . . . .
```